NASA Conference Publication 10112, Vol. II

NORTH AMERICAN FUZZY INFORMATION PROCESSING SOCIETY (NAFIPS '92)

James Villarreal, *Compiler*
NASA Lyndon B. Johnson Space Center
Houston, Texas

Proceedings of a conference sponsored by the
North American Fuzzy Information Processing Society
in cooperation with the National Aeronautics and Space
Administration, the Instituto Tecnologico de Morelia,
the Indian Society for Fuzzy Mathematics and Information
Processing, the Instituto Tecnologico de Estudios Superiores
de Monterrey, the International Fuzzy Theory and Systems,
and the Microelectronics and Computer Technology Corporation,
and held at the Melia Hotel Pasco de la Marina Sur, Marina
Vallarta, Puerto Vallarta, Mexico

December 15-17, 1992

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas
1992

NAFIPS '92, an international conference on fuzzy set theory and applications, is sponsored by NAFIPS, in cooperation with:

- National Aeronautics and Space Administration (NASA)
- instituto Tecnologico de Morella
- Indian Society for Fuzzy Mathematics and Information Processing(ISFUMIP)
- Instituto Tecnologico de Estudios Superiores de Monterrey (ITESM)
- International Fuzzy Systems Association (IFSA)
- Japan Society for Fuzzy Theory and Systems
- Microelectronics and Computer Technology Corporation (MCC)

Fuzzy set theory has led to a large number of diverse applications. Recently, interesting applications have been developed which involve the integration of fuzzy systems with adaptive processes such as neural networks and genetic algorithms. NAFIPS '92 will be directed toward the advancement, commercialization, and engineering development of these technologies.

The conference will consist of both plenary sessions and contributory sessions. The plenary sessions will be addressed by leading experts. Topics to be discussed at this conference include the following:

- Biomedical and Biochemical Issues
- Business and Decision Making
- Commercial Products and Tools
- Computer Systems and Information Processing
- Control Systems
- Decision Analysis
- Foundations and Mathematical Issues
- Genetic Algorithms
- Hardware
- Image Processing and Vision
- Neural Networks
- Optimization
- Path Planning
- Pattern Recognition
- Robotics

CONFERENCE CHAIRS

Honorary Chair
Lotfi Zadeh
University of California, Berkeley

General Chair
Robert N. Lea
NASA – Lyndon B. Johnson Space Center
Houston, TX 77058

Program Co-Chairs
James A. Villarreal
Software Technology Branch
Lyndon B. Johnson Space Center
NASA
Houston, TX 77058
Tel: (713)483-8076
EMAIL: JAMES@GOTHAMCITY.JSC.NASA.GOV

Salvador Guiterrez Martinez
Instituto Tecnologico de
Morelia,
Av. Tecnologico 1500
Col. Lomas de Santiaguito
58120 Morelia
Morelia, Michoacan, Mexico

PROGRAM COMMITTEE

Advisory Board
G. Klir
J. Keller
T. Yamakawa
Y. Jani

J. Bezdek
L. Hall
M. Sugeno
I. Turksen

R. Yager
M. Zemankova
S. K. Pal
M. Murphy

Committee
J. Aldridge
H. Berenji
J. Bezdek
P. Bonisonne
B. Buckles
J. Buckley
S. Chiu
T. Cleghorn
C. Copeland
A. deKorvin
J. Dockery
D. Dubois
T. Espy
W. Gillespie
M. Groff
S. Guiterrez-Martinez
L. Hall
T. Henson
J. Hoblitt
D. Hudson
Y. Jani
B. Jansen

C. Joslyn
C. Karr
J. Keller
J. Kowing
D. Kraft
V. Kreinovich
R. Krishnapuram
R. N. Lea
A. Leigh
D. Leo
M. Murphy
S. Nair
C. Negoita
S. K. Pal
A. Parlos
F. Petry
N. Pfluger
F. Pin
P. A. Ramamoothy
A. Ramer
E. Ruspini
E. Sanchez

B. Schott
H. Shehadeh
R. O. Shelton
S. Shenoi
M. Shipley
R. E. Smith
S. Mitra
T. Takagi
M. Valenzuela
J. A. Villarreal
L. Wang
P. Wang
H. Watanabe
W. Wee
J. Weiss
I. A. Willson
T. Whalen
R. Yager
J. Yao
J. Yen
H. Ying
M. Zemankova

____
(Initials)

ANALYTIC REVIEW FORM

Primary Record IPS # 161840

___ Document should not receive analytic treatment

SUBSIDIARY ADD

Document page range ___ to ___ New subsidiary # ___
Document page range ___ to ___ New subsidiary # ___
Document page range ___ to ___ New subsidiary # ___

SUBSIDIARY DELETE/CORRECTION

Subsidiary #___
(IPS#___________)

___ Delete

Reason: ___ limited technical content
___ no separate authorship
___ content dependent

___ Adjust paging
___ Other ______________________
____

Subsidiary #___
(IPS#___________)

___ Delete

Reason: ___ limited technical content
___ no separate authorship
___ content dependent

___ Adjust paging
___ Other ______________________
____

Subsidiary #___
(IPS#___________)

___ Delete

Reason: ___ limited technical content
___ no separate authorship
___ content dependent

___ Adjust paging
___ Other ______________________
____

TUTORIALS

Tutorials by leading experts will be provided on December 14, 1992.

8:00 – 9:40	**Introduction to Fuzzy Sets and Approximate Reasoning** RONALD R. YAGER, Iona College, New Rochelle, NY, USA
9:50 – 11:30	**Fuzzy Intelligent Information Systems** M. ZEMANKOVA, NATIONAL Science Foundation, Washington, DC, USA
11:30 – 12:30	**Lunch**
12:30 – 2:10	**Fuzzy Logic in Expert System and its Applications for IE/OR/MS** I.B. TURKSEN, University of Toronto, Toronto, ON, CANADA
2:20 4:00	**Fuzzy Control and its Applications** M. SUGENO, Tokyo Institute of Technology, Yokohama, JAPAN
4:10 – 5:50	**Fuzzy Hardware Design and its Applications** K. HIROTA, Hosei University, Tokyo, JAPAN

CONFERENCE

Tuesday, December 15, 1992

8:00	**Welcoming Remarks**
8:15–9:00	**Plenary Speech** PROFESSOR LOTFI ZADEH, University of California at Berkeley
9:00 - 12:00	**Parallel Sessions**

Fuzzy Theory & Systems

An Analysis of Possible Applications of Fuzzy Set Theory to the Credibility Theory
KRZYSZTOF OSTASZEWSKI, University of Louisville, Louisville, KY
WALDEMAR KARWOWSKI, University of Louisville, Louisville, KY

Estimations of Expectedness and Potential Surprise in Possibility Theory
HENRI PRADE, Universite Paul Sabatier, Toulouse Cedex, FRANCE
RONALD R. YAGER, Iona College, New Rochelle, NY

Comparison of Specificity and Information for Fuzzy Domains
ARTHUR RAMER, University of New South Wales, Kensington, AUSTRALIA

The Axiomatic Definition of a Linguistic Scale Fuzziness Degree, its Major Properties and Applications
ALEXANDER P. RYJOV, Soviet Association of Fuzzy Systems, Moscow, RUSSIA

How to Select Combination Operators for Fuzzy Expert Systems Using CRI
I.B. TURKSEN, University of Toronto, Toronto, Ontario, CANADA
Y. TIAN, University of Toronto, Toronto, Ontario, CANADA

Approximate Reasoning Using Terminological Models
JOHN YEN, Texas A&M University, College Station, TX
NITIN VAIDYA, Texas A&M University, College Station, TX

[illegible] Image Analysis

Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions
RAGHU KRISHNAPURAM, University of Missouri, Columbia, MO
JAMES M. KELLER, University of Missouri, Columbia, MO
YIBING MA, University of Missouri, Columbia, MO

A Fuzzy Clustering Algorithm to Detect Planar and Quadric Shapes
RAGHU KRISHNAPURAM, University of Missouri, Columbia, MO
HICHEM FRIGUI, University of Missouri, Columbia, MO
OLFA NASRAOUI, University of Missouri, Columbia, MO

A Fuzzy Measure Approach to Motion Frame Analysis for Scene Detection
ALBERT B. LEIGH, McDonnell Douglas Space Systems, Houston, TX
SANKAR K. PAL, Indian Statistical Institute, Calcutta, INDIA

Automatic Rule Generation for High-Level Vision
FRANK CHUNG-HOON RHEE, University of Missouri, Columbia, MO
RAGHU KRISHNAPURAM, University of Missouri, Columbia, MO

Encoding Spatial Images - A Fuzzy Set's Theory Approach
LESZEK M. SZTANDERA, University of Toledo, Toledo, OH

Image Segmentation Using LVQ Clustering Networks
ERIC CHEN-KUO TSAO, The University of West Florida, Pensacola, FL
JAMES C. BEZDEK, The University of West Florida, Pensacola, FL
NIKHIL R. PAL, The University of West Florida, Pensacola, FL

12:00 - 1:00 Lunch

1:00 - 3:30 Parallel Sessions

Adaptive Learning

A Neuro-Fuzzy Architecture for Real-Time Applications
P. A. RAMAMOORTHY, University of Cincinnati, Cincinnati, OH
SONG HUANG, University of Cincinnati, Cincinnati, OH

A Composite Self Tuning Strategy for Fuzzy Control of Dynamic Systems
C-Y SHIEH, University of Missouri, Columbia, MO
SATISH S. NAIR, University of Missouri, Columbia, MO

A Self-Learning Rule Base for Command Following in Dynamical Systems
WEI K. TSAI, University of California at Irvine, Irvine, CA
HON-MUN LEE, University of California at Irvine, Irvine, CA
ALEXANDER PARLOS, Texas A&M University, College Station, TX

Adaptive Defuzzification for Fuzzy Systems Modeling
RONALD R. YAGER, Iona College, New Rochelle, NY
DIMITAR P. FILEV, Iona College, New Rochelle, NY

Design Issues of a Reinforcement-Based Self-Learning Fuzzy Controller for Petrochemical Process Control
JOHN YEN, Texas A&M University, College Station, TX
HAOJIN WANG, Texas A&M University, College Station, TX
WALTER C. DAUGHERITY, Texas A&M University, College Station, TX

[illegible] Applications

Learning Characteristics of a Space Time Neural Network as a Tether Skiprope Observer
ROBERT N. LEA, NASA/Johnson Space Center, Houston, TX
JAMES A. VILLARREAL, NASA/Johnson Space Center, Houston, TX
JANI YASHVANT, Togal Infralogic Inc., Houston, TX
CHARLES COPELAND, Loral Space Systems, Houston, TX

Clustering of Tethered Satellite System Simulation Data by an Adaptive Neuro-Fuzzy Algorithm
SUNANDA MITRA, Texas Tech University, Lubbock, TX
SURYA PEMMARAJU, Texas Tech University, Lubbock, TX

Character Recognition Using a Neural Network Model with Fuzzy Representation
NASSRIN TAVAKOLI, University of North Carolina at Charlotte, Charlotte, NC
DAVID SENIW, University of North Carolina at Charlotte, Charlotte, NC

Designing a Fuzzy Scheduler for Hard Real-Time Systems
JOHN YEN, Texas A&M University, College Station, TX
JONATHAN LEE, Texas A&M University, College Station, TX
NATHAN PFLUGER, Texas A&M University, College Station, TX
SWAMI NATARAJAN, Texas A&M University, College Station, TX

WARP: Weight Associative Rule Processor A Dedicated VLSI Fuzzy Logic Megacell
ANDREA PAGNI, SGS-Thompson Microelectronics, Agrate Brianza (MI) ITALY
R. POLUZZI, SGS-Thompson Microelectronics, Agrate Brianza (MI) ITALY
G. G. RIZZOTTO, SGS-Thompson Microelectronics, Agrate Brianza (MI) ITALY

Wednesday, December 16, 1992

8:00 - 8:45 **Plenary Speech**
Piero Bonissone, "Fuzzy Logic Control: From Development to Deployment (with an Application to Aircraft Engine Control)"

8:45 - 10:45 **Parallel Sessions**

Decision Analysis

Evaluation of Fuzzy Inference Systems Using Fuzzy Least Squares
JOSEPH M. BARONE, Loki Software, Inc., Liberty Corner, NJ

A Model for Amalgamation in Group Decision Making
VINCENZO CUTELLO, Consorzio per la Ricerca sulla Microelettronica del Mezzogiorno, Catania, ITALY
JAVIER MONTERO, Complutense University, Madrid, Spain

Fuzzy Forecasting and Decision Making in Short Dynamic Time Series
EFIN JA. KARPOVSKY, Odessa Institute of National Economy, Odessa, UKRAINE

Decision Analysis With Approximate Probabilities
THOMAS WHALEN, Georgia State University, Atlanta, GA

Applications

Distributed Traffic Signal Control Using Fuzzy Logic
STEPHEN CHIU, Rockwell International Science Center,
Thousand Oaks, CA

Intelligent Virtual Reality in the Setting of Fuzzy Sets
JOHN T. DOCKERY, George Mason University, Fairfax, VA
DAVID LITTMAN, George Mason University, Fairfax, VA

Comparison of Crisp and Fuzzy Character Networks in Handwritten Word Recognition
PAUL GADER, University of Missouri, Columbia, MO
MAGDI MOHAMED, University of Missouri, Columbia, MO
JUNG-HSIEN CHIANG, University of Missouri, Columbia, MO

Fuzzy Neural Network Methodology Applied to Medical Diagnosis
MARIAN B. GORZALCZANY, Technical University of Kielce, Kielce, POLAND
MARY DEUTSCH-MCLEISH, University of Guelph, Guelph, Ontario, CANADA

11:00 - 12:00 Parallel Sessions

Decision Analysis

An Experimental Methodology for a Fuzzy Set Preference Model
I.B. TURKSEN, University of Toronto, Toronto, ON, CANADA
IAN A. WILLSON, University of Toronto, Toronto, ON, CANADA

A Fuzzy Set Preference Model for Market Share Analysis
I.B. TURKSEN, University of Toronto, Toronto, ON, CANADA
IAN A. WILLSON, University of Toronto, Toronto, ON, CANADA

Knowledge Representations

Information Compression in the Context Model
JORG GEBHARDT, Technical University of Braunschweig, Braunschweig, GERMANY
RUDOLF KRUSE, Technical University of Braunschweig, Braunschweig, GERMANY
DETLEF NAUCK, Technical University of Braunschweig, Braunschweig, GERMANY

Fuzzy Knowledge Base Construction Through Belief Networks Based on Lukasiewicz Logic
FELIPE LARA-ROSANO, Universidad Nacional Autonoma de Mexico, Mexico DF, MEXICO

12:00 - 1:00 Lunch

1:00 - 3:30 Parallel Sessions

Control Systems

Intelligent Fuzzy Controller for Event-Driven Real Time Systems
JANOS GRANTNER, University of Minnesota, Minneapolis, MN
MAREK PATYRA, University of Minnesota, Minneapolis, MN
MARIAN S. STACHOWICZ, University of Minnesota, Minneapolis, MN

Fuzzy Coordinator in Control Problems
A. RUEDA, University of Manitoba, Winnipeg, Manitoba, CANADA
W. PEDRYCZ, University of Manitoba, Winnipeg, Manitoba, CANADA

Tuning a Fuzzy Controller Using Quadratic Response Surfaces
BRIAN SCHOTT, Georgia State University, Atlanta, GA
THOMAS WHALEN, Georgia State University, Atlanta, GA

The Cognitive Bases for the Design of a New Class of Fuzzy Logic Controllers: The Clearness Transformation Fuzzy Logic Controller
LABIB SULTAN, York University, Toronto, Ontario, CANADA
TALIB JANABI, Mentalogic Systems Inc., Markham,
Ontario, CANADA

A Fuzzy Control Design Case: The Fuzzy PLL
H.N. TEODORESCU, Polytechnic Institute of Iasi, ROMANIA
I. BOGDAN, Polytechnic Institute of Iasi, ROMANIA

Adaptive Learning

Adding Dynamic Rules to Self-Organizing Fuzzy Systems
CATALIN V. BUHUSI, Romanian Academy, Calea Copou, Iasi, ROMANIA

Fuzzy Learning Under and About an Unfamiliar Fuzzy Teacher
BELUR V. DASARATHY, Dynetics, Huntsville, AL

Some Problems with the Design of Self-Learning Management Systems
ZINY FLIKOP, NYNEX Science and Technology, Inc.,
White Plains, NY

A Neural Fuzzy Controller Learning by Fuzzy Error Propagation
DETLEF NAUCK, Technical University of Braunschweig, Braunschweig, GERMANY
RUDOLF KRUSE, Technical University of Braunschweig, Braunschweig, GERMANY

Thursday December 17, 1992

8:00 - 10:00 Parallel Sessions

Decision Analysis

Determining Rules for Closing Customer Service Centers: A Public Utility Company's Fuzzy Decision
ANDRE DEKORVIN, University of Houston - Downtown, Houston, TX
MARGARET F. SHIPLEY, University of Houston - Downtown,
Houston, TX
ROBERT N. LEA, NASA/Johnson Space Center, Houston, TX

Fuzzy Simulation in Concurrent Engineering
A. KRASLAWSKI, Lappeenranta University of Technology, Lappeenranta, FINLAND
L. NYSTROM, Lappeenranta University of Technology, Lappeenranta, FINLAND

Inverse Problems: Fuzzy Representation of Uncertainty Generates a Regularization
V. KREINOVICH, University of Texas at El Paso, El Paso, TX
CHING-CHUANG CHANG, University of Texas at El Paso, El Paso, TX
L. REZNIK, Victoria University of Technology, MMC Melbourne,
VIC 3000, AUSTRALIA
G. N. SOLOPCHENKO, St. Petersburg Technical University, St. Petersburg, RUSSIA

Quantification of Human Responses
RALPH C. STEINLAGE, University of Dayton, Dayton, OH
T. E. GANTNER, University of Dayton, Dayton, OH
P. Y. W. LIM, Boise Cascade R&D, Portland, OR

Fuzzy Theory & Systems

Non-Scalar Uncertainty
SALVADOR GUTIERREZ–MARTINEZ, Instituto Tecnologico de Morelia, Morelia, MEXICO

Comparison Between the Performance of Two Classes of Fuzzy Controllers
TALIB H. JANABI, Mentalogic Systems Inc., Markham, Ontario, CANADA
L.H. SULTAN, York University, Toronto, Ontario, CANADA

Possibilistic Measurement and Set Statistics
CLIFF JOSLYN, SUNY-Binghamton, Portland, ME

The Fusion of Information via Fuzzy Integration
JIM KELLER, University of Missouri, Columbia, MO
HOSSEIN TAHANI, University of Missouri, Columbia, MO

10:15 - 11:45 Parallel Sessions

Database Management

On the Evaluation of Fuzzy Quantified Queries in a Database Management System
PATRICK BOSC, IRISA/ENSSAT, Lannion, Cedex, FRANCE
OLIVIER PIVERT, IRISA/ENSSAT, Lannion, Cedex, FRANCE

A Fuzzy Case Based Reasoning Tool for Model Based Approach to Rocket Engine Health Monitoring
SRINIVAS KROVVIDY, University of Cincinnati, Cincinnati, OH
ADAM NOLAN, University of Cincinnati, Cincinnati, OH
YONG LIN HU, University of Cincinnati, Cincinnati, OH
WILLIAM G. WEE, University of Cincinnati, Cincinnati, OH

A High Performance, Ad-Hoc Fuzzy Query Processing System for Relational Databases
W.H. MANSFIELD, Bellcore, Cambridge, MA, USA
ROBERT M. FLEISCHMAN, BBN, Cambridge, MA, USA

Genetic Algorithms/Optimization

Genetic Algorithms in Adaptive Fuzzy Control
C. LUCAS KARR, U. S. Department of Interior Bureau of Mines, Tuscaloosa, AL

A Genetic Algorithms Approach for Altering the Membership Functions in Fuzzy Logic Controllers
HANA SHEHADEH, LinCom Corporation, Houston, TX
ROBERT N. LEA, NASA/Johnson Space Center, Houston, TX

Fuzzy Multiple Linear Regression - A Computational Approach
C.H. JUANG, Clemson University, Clemson, SC
X.H. HUANG, Clemson University, Clemson, SC
J.W. FLEMING, Clemson University, Clemson, SC

12:00 - 1:00 Lunch

1:00 - 4:30 Parallel Sessions

Neural Networks

Incorporation of Varying Types of Temporal Data in a Neural Network
M. E. COHEN, California State University, Fresno, CA
D.L. HUDSON, California State University, Fresno, CA

Fuzzy Operators and Cyclic Behaviour in Formal Neural Networks
E. LABOS, Semmelweis University Medical School, Budapest, HUNGARY
A. V. HOLDEN, The University of Leeds, Leeds, UK
J. LACZKO, Ludwig Maximilien University, Munchen, GERMANY
A. S. LABOS, Semmelweis University Medical School, Budapest, HUNGARY

Neural Networks: A Simulation Technique Under Uncertainty Conditions
LUISA MCALLISTER, Moravian College, Bethlehem, PA

Incomplete Fuzzy Data Processing Using Artificial Neural Network
MAREK J. PATYRA, University of Minnesota, Duluth, MN

Stochastic Architecture for Hopfield Neural Nets
SANDY PAVEL, Polytechnical Institute of Iasi, Iasi, ROMANIA

Hierarchical Model of Matching
W. PEDRYCZ, University of Manitoba, Winnipeg, Manitoba, CANADA
EUGENE ROVENTA, York University, Toronto, Ontario, CANADA

A Conjugate Gradients/Trust Regions Algorithm for Training Multilayer Perceptrons for Nonlinear Mapping
RAGHAVENDRA K. MADYASTHA, Rice University, Houston, TX
BEHNAAM AAZHANG, Rice University, Houston, TX
TROY F. HENSON, IBM Corporation, Houston, TX
WENDY L. HUXHOLD, IBM Corporation, Houston, TX

Fuzzy Theory & Systems

On Probability-Possibility Transformations
GEORGE KLIR, State University of New York, Binghamton, NY
BEHZAD PARVIZ, California State University, Los Angeles, CA

Inference in Fuzzy Rule with Conflicting Evidence
LASZLO T. KOCZY, Technical University of Budapest, Budapest, HUNGARY

Gaussian Membership Functions are Most Adequate in Representing Uncertainty in Measurements
V. KREINOVICH, University of Texas at El Paso, El Paso, TX
C. QUINTANA, University of Michigan at Ann Arbor, Ann Arbor, MI
L. REZNIK, Victoria University of Technology, MMC Melbourne,
VIC 3000, AUSTRALIA

Applying the Metric Truth Approach to Fuzzified Automated Reasoning
VESA A. NISKANEN, University of Helsinki, Helsinki, FINLAND

Life Insurance Risk Assessment Using a Fuzzy Logic Expert System
L. A. CARRENO, Togai InfraLogic, Houston, TX
R. A. STEEL, Togai InfraLogic, Houston, TX

CONTENTS

An Analysis of Possible Applications of Fuzzy Set Theory to the Actuarial Credibility Theory 1

Estimations of Expectedness and Potential Surprise in Possibility Theory 7

Comparison of Specificity and Information for Fuzzy Domains 14

The Axiomatic Definition of a Linguistic Scale Fuzziness Degree, Its Major Properties and Applications 21

How to Select Combination Operators for Fuzzy Expert Systems Using CRI 29

Approximate Reasoning Using Terminological Models 39

Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions 49

A Fuzzy Clustering Algorithm to Detect Planar and Quadric Shapes 59

A Fuzzy Measure Approach to Motion Frame Analysis for Scene Detection 69

Automatic Rule Generation for High-Level Vision 81

Encoding Spatial Images - A Fuzzy Set Theory Approach 89

Image Segmentation Using LVQ Clustering Networks 98

A Neuro-Fuzzy Architecture for Real-Time Applications 108

A Composite Self Tuning Strategy for Fuzzy Control of Dynamic Systems 118

A Self-Learning Rule Base for Command Following in Dynamical Systems 125

Adaptive Defuzzification for Fuzzy Systems Modeling 135

Design Issues of a Reinforcement-Based Self-Learning Fuzzy Controller for Petrochemical Process Control 143

Learning Characteristics of a Space-Time Neural Network as a Tether "Skiprope Observer" 154

Clustering of Tethered Satellite System Simulation Data by an Adaptive Neuro-Fuzzy Algorithm 166

Character Recognition Using a Neural Network Model with Fuzzy Representation 175

Designing a Fuzzy Scheduler for Hard Real-Time Systems 185

WARP: Weight Associative Rule Processor, A Dedicated VLSI Fuzzy Logic Megacell 195

Evaluation of Fuzzy Inference Systems Using Fuzzy Least Squares 205

A Model for Amalgamation in Group Decision Making 215

Fuzzy Forecasting and Decision Making in Short Dynamic Time Series 224

Decision Analysis With Aproximate Probabilities 229

Distributed Traffic Signal Control Using Fuzzy Logic 239

Intelligent Virtual Reality in the Setting of Fuzzy Sets 248

Comparison of Crisp and Fuzzy Character Networks in Handwritten Word Recognition 257

Fuzzy Neural Network Methodology Applied to Medical Diagnosis 266

An Experimental Methodology for a Fuzzy Set Preference Model 276

A Fuzzy Set Preference Model for Market Share Analysis 286

Information Compression in the Context Model 296

Fuzzy Knowledge Base Construction Through Belief Networks Based on Lukasiewicz Logic 304

Intelligent Fuzzy Controller for Event-Driven Real Time Systems 312

Fuzzy Coordinator in Control Problems 322

Tuning a Fuzzy Controller Using Quadratic Response Surfaces 330

The Cognitive Bases fcr the Design of a New Class of Fuzzy Logic Controllers: The Clearness Transformation Fuzzy Logic Controller 340

A Fuzzy Control Design Case: The Fuzzy PLL 350

Adding Dynamic Rules to Self-Organizing Fuzzy Systems 360 1

Fuzzy Learning Under and About an Unfamiliar Fuzzy Teacher 368 2

Some Problems With the Design of Self-Learning Management Systems 378 3

A Neural Fuzzy Controller Learning by Fuzzy Error Propagation 388 4

Determining Rules for Closing Customer Service Centers: A Public Utility Company's Fuzzy Decision 398 5

Fuzzy Simulation in Concurrent Engineering 408 6

Inverse Problems: Fuzzy Representation of Uncertainty Generates a Regularization 418 7

Quantification of Human Responses 427 8

Non-Scalar Uncertainty 437 9

Comparison Between the Performance of Two Classes of Fuzzy Controllers 449 10

Possibilistic Measurement and Set Statistics 458 11

The Fusion of Information Via Fuzzy Integration 468 12

On the Evaluation of Fuzzy Quantified Queries in a Database Management System 478 13

A Fuzzy Case Based Reasoning Tool for Model Based Approach to Rocket Engine Health Monitoring 488 14

A High Performance, Ad-Hoc, Fuzzy Query Processing System for Relational Databases 496 15

Genetic Algorithms in Adaptive Fuzzy Control 506 16

A Genetic Algorithms Approach for Altering the Membership Functions in Fuzzy Logic Controllers 515 17

Fuzzy Multiple Linear Regresssion - A Computational Approach 524 18

Incorporation of Varying Types of Temporal Data in a Neural Network 535 19

Fuzzy Operators and Cyclic Behavior in Formal Neuronal Networks 545 20

Neural Networks: A Simulation Technique Under Uncertainty Conditions 555 21

Incomplete Fuzzy Data Processing System Using Artificial Neural Network 585 22

Stochastic Architecture for Hopfield Neural Nets .... 575 -23

Hierarchical Model of Matching .... 581 -24

A Conjugate Gradients/Trust Regions Algorithm for Training Multilayer Perceptrons for Nonlinear Mapping .... 589 -25

On Probability-Possibility Transformations .... 598 -26

Inference in Fuzzy Rule Bases with Conflicting Evidence .... 608 -27

Gaussian Membership Functions Are Most Adequate in Representing Uncertainty in Measurements .... 618 -28

Applying the Metric Truth Approach to Fuzzified Automated Reasoning .... 625 -0

Life Insurance Risk Assessment Using a Fuzzy Logic Expert System.......... 627-29

Author Index A-1

Adding Dynamic Rules to Self-Organizing Fuzzy Systems

Catalin V. Buhusi
Romanian Academy, Institute for Computer Science,
Calea Copou nr.22A, IASI 6600, ROMANIA

Abstract

This paper develops a Dynamic Self-Organizing Fuzzy System (DSOFS) capable of adding, removing and/or adapting the fuzzy rules and the fuzzy reference sets. The DSOFS background consists in a self-organizing neural structure with neuron relocation features which will develop a map of the input-output behaviour. The relocation algorithm extends the topological ordering concept. Fuzzy rules (neurons) are dynamically added or released while the neural structure learns the pattern. The DSOFS advantages are the automatic synthesis and the possibility of parallel implementation. One could remark a high adaptation speed and the reduced number of neurons needed in order to keep errors under some limits. The computer simulation results in a nonlinear systems modelling application are shown.

keywords: fuzzy systems, neural networks, neuron relocation, Kohonen self-organizing procedure, LMS procedure, feature map, basin of attraction, lateral feed-back.

I. Introduction

The promising link between the fuzzy reasoning and the massively parallel calculus, i.e. fuzzy neural networks, became an important topic of the fuzzy systems research during the last years.

Learning on membership functions and the fuzzy rules are major problems in synthesizing a fuzzy system. In this paper we are interested in the automatic synthesis of reference sets and fuzzy rules. One of the classes of fuzzy systems which gives a solution to these problems is based on self-organizing neural structures which map the desired topological relations between the fuzzy system input and output. Some solutions are briefly discussed in section III.

This paper presents a Dynamic Self-Organizing Fuzzy System (DSOFS) capable of adding, adapting and/or removing the fuzzy rules and the reference fuzzy sets. The fuzzy system synthesis is based on a modifiable adaptive neural network using a neuron relocation algorithm as a learning method. This algorithm extends the topological ordering concept [4,7]. In the adaptation process neurons are added and/or disposed while learning the pattern. This is the neural equivalent of modifying the fuzzy system rules. The relocation algorithm supposes for every fuzzy rule (neuron) a basin of attraction as a base for the fuzzy reference sets construction.

II. The Dynamic Self-Organizing Fuzzy System Definition

In order to fix the ideas we will denote by R^n the input universe of discourse and by R^m the output universe of discourse, where n and m are fixed integers.

The DSOFS input and output are vectors in Rx...xR. We will denote such vectors as X, Y, or $\{x_1, x_2, ..., x_n\}$, $\{y_1, y_2, ..., y_m\}$.

The rules of the DSOFS have the

following form:

$$\text{if } X \text{ is } X^r \text{ then } Y \text{ is } Y^r \text{ with } b^r \qquad (1)$$

where $X \in R^n$ is the input vector; $Y \in R^m$ is the output vector; $X^r \in R^n$ is the input reference vector for the rule number r; $Y^r \in R^m$ is the output reference vector for the rule number r; b^r is the basin of attraction of the rule number r, $b^r \in R^+$.

The truth degree w^r of any rule r is given by:

$$w^r = F[b^r](\, d(X, X^r)\,) \qquad (2)$$

where b^r is the basin of attraction of the rule number r; d(*,*) is the Euclidian distance; $F[b^r](*)$ is a family of functions of parameter b^r such that:

(i) $F[b^r] : R^+ \rightarrow R$, $\forall b^r \in R^+$;
(ii) $F[b^r]$ is monotone decreasing $\forall b^r \in R^+$;
(iii) $F[b^r](0) = 1$, $\forall b^r \in R^+$;
(iv) if $b_i > b_j$ then $F[b_i](z) > F[b_j](z)$, $\forall z \in R^+$ and $b_i, b_j \in R^+$

The fuzzy system output Y is computed via:

$$y_i = \sum_{r=1}^{N} w^r * y^r_i \Big/ \sum_{r=1}^{N} w^r \;,\; i=1..m \qquad (3)$$

where N is the number of rules; w^r, y^r have their previous meanings.

III. The Dynamic Self-Organizing Fuzzy System Synthesis

A fuzzy system synthesis has to solve two problems: construction of the fuzzyfier, i.e. obtaining the reference fuzzy sets of the system, for both input and output, and construction of the fuzzy rules. These problems find particular solutions when fuzzy reasoning is linked on neural networks, and especially on self-organizing neural networks.

Yamaguchi et al. [13] proposed unsupervised learning the membership functions using the Learning Vector Quantization procedure [8], and the if-then-rule part using Bidirectional Associative Memories (BAMs) to show the relationships interpreted from fuzzy rules. Another approach made by Takagi and Hayashi [11] is using two kinds of neural networks, for the membership functions and for the fuzzy system output, networks whose adaptation and optimization are made by clustering algorithms. Bezdek proposed recently [1] a fuzzy Kohonen self-organizing system, an approach linking the Kohonen self-organizing procedure and fuzzy systems. Such a link was also proposed in [2].

One of the backdraws of the Kohonen self-organizing procedure, and of others clustering algorithms as well, is that the first engineering decision to be made is how many nodes should be used.

The Dynamic Self-Organizing Fuzzy Systems solve all these problems, based on a self-organizing neural network with neuron relocation features. Through the "learning" stage, the fuzzy rules are changed by adapting both input and output reference vectors and their basins of attraction. If necessary, new rules will be added and/or the old ones removed. The output of the fuzzy system will be therefore refined through an adaptation algorithm. This adaptation is made such that the energy of the difference between the DSOFS output and a desired sequence of outputs is minimized. The used adaptation algorithm is the well-known Least Mean Square (LMS) algorithm for adaptive linear combiners [12].

III.1. The Neuron Relocation Self-Organizing Procedure

A neural network implements the behaviour rules in the net weights. The

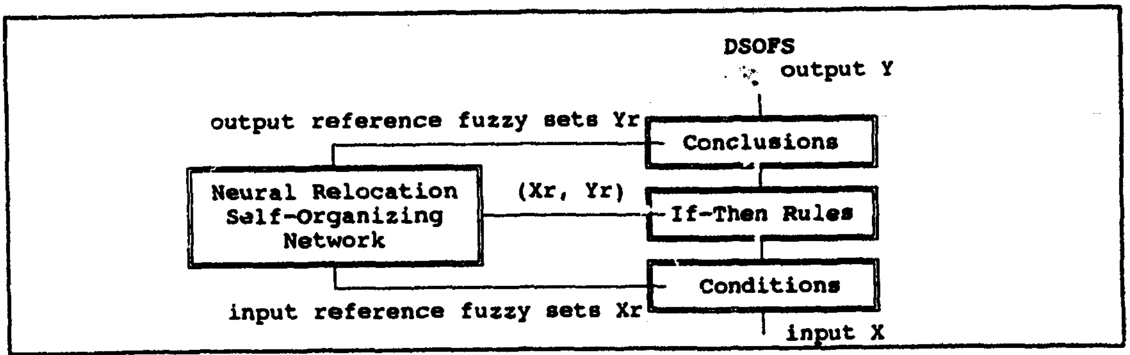


Fig.1 The DSOFS and the Neural Relocation Network

adaptation algorithm proposed in [4,7] by Kohonen is based on lateral feed-back concept. Networks using this biologically motivated process will behave such that network outputs form clusters around the excitation input local maxima. Such a neural structure supposes constant number of neurons, free of the information conveyed by the pattern, [4], in the sense of the topological distribution. Thus, a network with a given number of neurons could obviously hold less information for a nonuniform input distribution in opposition to a uniform one.

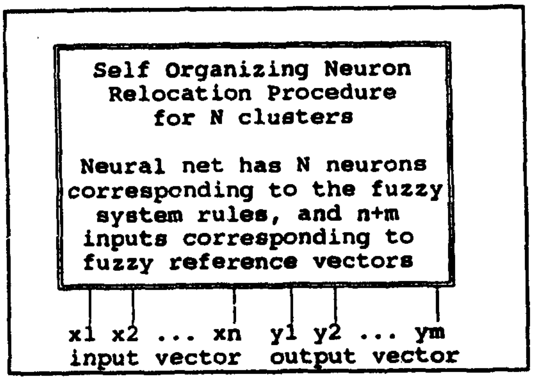


Fig.2 The Self-Organizing Neuron Relocation Procedure (Block Scheme)

On the contrary, a dynamical neural structure [3] could be distinguished by a neuron adding-releasing character as an aspect of the pattern novelty features. The DSOFS input-output mapping will be obtained via a self-organizing neuron relocation procedure which adapts the number of neurons (fuzzy rules), the weights and the neurons basin of attraction, forming clusters around the best matching neuron (fuzzy rule). We further propose a clustering algorithm which increases the adaptation speed and adapts the required number of fuzzy rules.

We will work with a neural net containing a variable number of neurons, equal to the number of fuzzy rules of the DSOFS, i.e. N. The behaviour of the DSOFS consists in the pairs $\{X^r, Y^r\}$ and in their basin of attraction b^r. In the adaptation stage the input-output pair $\{X, Y\}$ will feed the neural network which will map the input-output behaviour of the DSOFS in the net weights. These weights are the reference vectors of the fuzzy system $\{X^r, Y^r\}$.

In the followings we will use the Euclidian distance as a measure of similarity.

The relocation algorithm is based on a dynamical neuron allocation in terms of the input distribution specificity. Therefore, we propose the insertion of a new rule, i.e. of a new neuron, while the input is outside the basin of attraction of every rule in the actual set of rules. A rule will be removed if it is inside the basin of attraction of another rule. If none of the above, the rule adaptation process continues in order to build the reference fuzzy vectors, i.e. the neural feature map.

We will denote by N the variable number of neurons, each one represented by the radius of the basin of attraction b^r, varying between two fixed limits B_{min} and B_{max}, and n+m connection weights h_{ir} between the input i, i=1..n+m, and the r-th neuron. In fact, every ordered set $\{h_{1r}, ..., h_{n+m,r}\}$ may be regarded as a kind of image that shall be matched against the input vector $\{x_1, ...x_n, y_1, ..., y_m\}$

a. Initialize Structure

Initialize N with N_0, b^r with B_{min}, h_{ir} with small random values.

b. Develop Background

Train the network in order to produce the feature map formation via the successive presentation of some n+m samples from the pattern, breaking the process before the convergence phase, [4]. This step gives a background to the network for future adaptation and avoids an insertion explosion (see section IV).

c. Present New Sample

Present input vector Z={X, Y} and compute the Euclidian distance to the N neurons:

$$d(j) = \mathrm{sqrt}\left(\sum_{i=1}^{n+m} (h_{ij}-z_{ij})^2\right), j=1..N \quad (4)$$

d. Select Best Matching Neuron

Select the neuron k such that:

$$d(k) = \mathbf{min}\ \{d(j)\}, j=1..N \quad (5)$$

e. Insert New Neuron

If $d(k) > B_{max}$ then insert the p=N+1 -th neuron such that:

$$h_{ip} = z_i, i=1..n+m \quad (6)$$

Update the number of neurons N, and repeat by going to step c.

f. Adapt Network

If $b^k < d(k) \leq B_{max}$ then adapt the network in order to yield the characteristic feature map (7). Then repeat by going to step c.

The weights will assume new values in the process formally specified by:

$$\partial h_{ij}/\partial t = f(t)*(z_i-h_{ij}), i=1..n+m, j \in Nb(k,t)$$
$$\partial h_{ij}/\partial t = 0, \text{otherwise} \quad (7)$$

where k denotes the neuron with the best matching between the input {X, Y} and the weights; Nb(k,t) denotes a time decreasing neighborhood of the k-th neuron; f(t) denotes a slowly decreasing function of time, determined by experience.

g. Release Neuron

If $d(k) \leq b^k$ then increment the basin of attraction of the best matching neuron:

$$b^k = b^k + \epsilon, \epsilon > 0 \quad (8)$$

Verify if some neuron is inside the basin of attraction of the k-th neuron's and if

$$\mathrm{sqrt}\left(\sum_{i=1}^{n+m} (z^k_i - z^p_i)^2\right) < b^k, 1 \leq p \leq N, p \neq k \quad (9)$$

remove neuron p and update the number of neurons. Repeat by going to step c.

•••

This algorithm will provide the fuzzy rules, i.e. the neurons with their basin of attraction, and the fuzzy reference vectors, i.e. the pairs $\{X^r, Y^r\}$ consisting in the weights h_{ir} of the network.

III.2. The DSOFS Refined Synthesis via LMS Adaptation Procedure

The synthesis of the fuzzy system via the self-organizing neuron relocation gives only a mean estimation of the pairs $\{X^r, Y^r\}$. The reference fuzzy sets X^r and Y^r may be considered satisfactory and the computer simulations showed that the basic properties of the input-output behaviour are well preserved by neural learning. A great

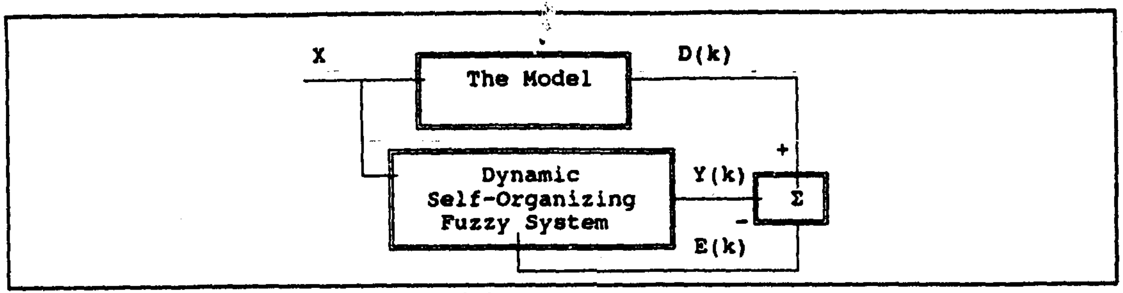


Fig.3 The LMS Adaptation (Block Scheme)

improvement may be obtained by adapting the output reference fuzzy sets such that the fuzzy system output will reach the desired output [2]. A classical adaptation algorithm is the LMS procedure. The block scheme of the LMS adaptation is depicted in Fig.3.

Suppose that we have at the moment k, $D(k) \in R^m$ the desired output vector, and $Y(k) \in R^m$ the actual fuzzy system output.

The error vector E(k) is given by:

$$E(k) = D(k) - Y(k) \qquad (10)$$

The LMS adaptation rules are:

$$y_i(k+1) = y_i(k) + \mu * e_i(k) * \alpha^r(k), \quad i=1..m,\ r=1..N \qquad (11)$$

where μ is the adaptation factor; k is the iteration number; $\alpha^r(k)$ is given by:

$$\alpha^r(k) = w^r(k) \,/\, \sum_{r=1}^{N} w^r(k), \quad r=1..N \qquad (12)$$

where $w^r(k)$ is the truth degree of the fuzzy rule number r.

IV. About The Neuron Relocation Algorithm

The neural relocation algorithm presented above may be matched against other clustering algorithms.

The complexity level is the same as Kohonen's procedure. Nevertheless, after the developing background stage, in the neural relocation algorithm the "learning by insertion" method will reduce the time needed in order to adapt the network, because it is worth copying the input-output behaviour than adapt the weights. If a fuzzy rule is no longer needed (i.e. the rule is inside the basin of attraction of another rule) it will be removed, so the complexity diminish.

Of course, in the above algorithm could be inserted a number of neurons equal to the number of iterations. This can be balanced by changing the maximum basin radius B_{max}. The maximum radius B_{max} is important in both minimizing the errors and the number of neurons. The minimum basin radius B_{min} may be chosen to be null or a small positive value, contributing only at the convergence time.

The simulation results showed the background developing stage to be very important in the neuron cost. If this stage is overstepped, the adaptation will evolve such that the initial iterations will add random neurons and it will take some time to remove some wrong positioned ones. This stage will form a basis on which the relocation features act well. The initial number of neurons N_0 has a similar importance in the neuron cost. It can be null, but this is not recommended.

The Grossberg ART net [5] has also the feature of inserting neurons based on a parameter called vigilance. The same problems (adding too many nodes or having too little discrimination) arise. In opposition to ART, the proposed model uses a parameter controlling the insertion of neurons ,i.e. B_{max}, and also a number of N parameters controlling the adaptation and deletion of the neurons (fuzzy rules), i.e. b^r, r=1..N. The B_{max} parameter is fixed as the vigilance is in the ART model, but the b^r are adaptive parameters.

Our simulations showed the time needed to adapt the net to be at least 25% lower than the Kohonen self-organizing method, and the number of neurons needed in order to represent the input-output behaviour diminished at about half. This can be explained by the "insertion" effect and the radius of attraction which can substitute neurons.

V. An application: Nonlinear System Modelling

We have applied DSOFS in a nonlinear system modeling application. This problem it is really suited to the DSOFS. It involves a model and a fuzzy system which will "learn" the behaviour of the model. The input and the output of the DSOFS are suited as the same dimensions as the model. In the first step of the synthesis a neural network "learns" the behaviour of the model. This phase will give us the reference vectors, the number of fuzzy rules and their basins of attraction. The output reference vectors will be adapted through LMS procedure in the second step of the synthesis in order to obtain a better resemblance to the model.

In the computer simulations that we further present we have used a multi-input single output nonlinear model with the input-output behaviour depicted in Figure 4, described by:

$$y = \exp(-x_1^2-x_2^2), \quad x_1 \in [-2, 2], \; x_2 \in [-2, 2] \qquad (13)$$

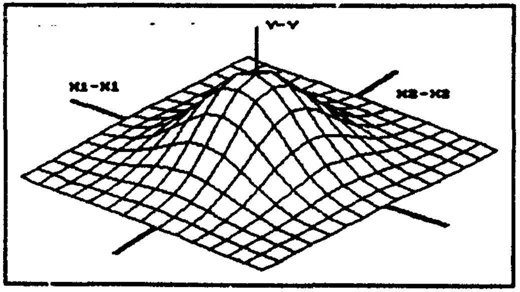


Fig.4 The Nonlinear System Output

The DSOFS has n=2, m=1 and rules of the form:

if X is X^r then y is y^r with b^r, r=1..N (14)

where $X = \{x_1, x_2\}$, $X \in R^2$; $y \in R$; $b^r \in R^+$.

In the self-organizing step of the synthesis we have used a neural network of inputs x_1, x_2 and y, with $N_0=40$ neurons (fuzzy rules). We have stopped the preliminary adaptation process before the 1500-th iteration (developing background step). Afterwards, we have successively presented samples from the pattern according to the neural relocation algorithm proposed above.

The similarity measure that we have used in our computer simulations was of the following form:

$$d(\{x_1. x_2, y\}, \{h_{1r}, h_{2r}, h_{3r}\}) = a^2*(x_1-h_{1r})^2+a^2*(x_2-h_{2r})^2+b^2*(y-h_{3r})^2 \qquad (15)$$

where $b>a$, $a,b \in R^+$.

The simulations results also showed that the network have the tendency to add less neurons while the process continues, as an effect of increasing the basins of attraction of the fuzzy rules up to the maximum basin radius. In Fig.5 it is depicted an example of the distribution of the rules in a step of the neural relocation self-organizing procedure. Every rule r is represented by a circle with the center (x_1^r, x_2^r) and the radius b^r.

After the neuron relocation self-organizing

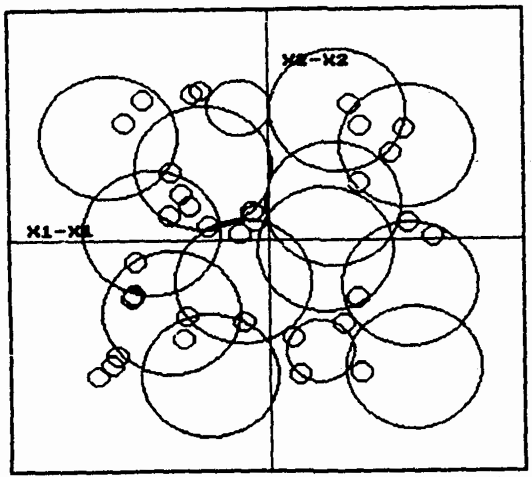


Fig.5 The Spatial Distribution of the Basins of Attraction

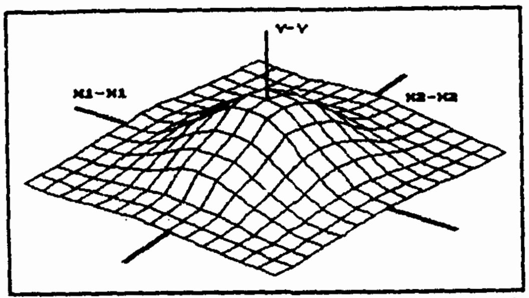


Fig.6 The DSOFS Output after the Self-Organizing Procedure

procedure we have obtained N=51 rules $\{x^r_1, x^r_2, y^r\}$ consisting in the network weights $\{h_{1r}, h_{2r}, h_{3r}\}$,i=1..N. In Fig.6 it is depicted the output of the fuzzy system after 5000 iteration of the self-organizing procedure. We can note the well topological resemblance to the model (including the symmetries).

The truth degree of the rule r was computed by:

$$w^r(x_1,x_2) = g_1(x_1-x^r_1)*g_2(x_2-x^r_2)*g_3(b^r) \quad (16)$$

where g_1, g_2, g_3 are gaussian-like functions.

These vectors became a background for

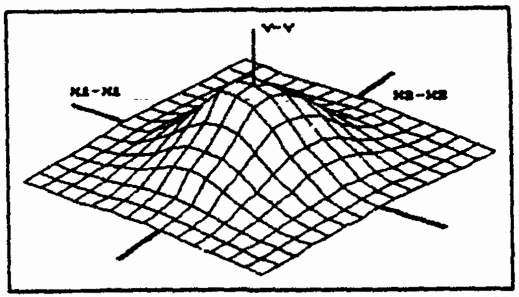


Fig.7 The DSOFS output after the LMS Adaptation Procedure

the second step: LMS adaptation of the output. In the LMS adaptation procedure we redraw the output reference fuzzy sets y^r in order to obtain better results. In Fig.7 it is depicted the output of the fuzzy system after 1000 iterations of the LMS procedure.

VI. Conclusions

The Dynamic Self-Organizing Fuzzy Systems have some major advantages based on rules adding/removing features and the reference fuzzy sets adaptation: [1] automatic synthesis based on neuron relocation self-organizing procedure and the LMS adaptation; [2] the possibility of parallel implementation

The DSOFS background consists in a self-organizing neural network with neuron relocation features. The neural equivalent of adding/removing rules is relocation of the neurons. According to the proposed clustering algorithm, neurons (fuzzy rules) are relocated and the fuzzy reference sets for both input and output are adapted in order to develop feature map formation. One could remark a higher adaptation speed and the reduced numbers of neurons in comparison with Kohonen's self-organizing model.

These advantages impose them in the problems involving modelling, automatic fuzzy system synthesis and adaptation. They can be both used in the developing stage of other fuzzy systems and in self-sustained applications.

References

[1] Bezdek,J.C., Tsao,E.C.K, Pal,N.R; *Fuzzy Kohonen Clustering Networks*, FUZ-IEEE'92, San Diego California, 1992;
[2] Buhusi,C.V., Chelaru,M., Dragoi, V.; *Self-Organizing Fuzzy Systems*, Int. Fuzzy Syst. and Artif. Intell. Symp., Iasi, Romania, 1991;
[3] Buhusi,C.V., Dragoi,V.; *Neural Structures with Neuron Relocation Features*, INFO-IASI Computer Science Symp. Proc., 1991;
[4] Kohonen,T.; *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, 1984;
[5] Grossberg,S.; *The Adaptive Brain*, Elsevier/North-Holland, Amsterdam, 1986;
[6] Kang,G., Sugeno,M.; *Fuzzy Modelling*, Trans. of SICE, 1987;
[7] Kohonen,T.; *Internal Representation and Associative Memory*, Elsevier Science Publishers, 1990;
[8] Kohonen,T.; *Learning Vector Quantization for Pattern Recognition*, Tech. Report of Helsinki Univ., 1986;
[9] Pedrycz,W.; *Fuzzy Control and Fuzzy Systems*, Research Studies Ltd., John Wiley & Sons, 1989;
[10] Pedrycz,W.; *Tutorial on Fuzzy Systems for Pattern Recognition*, IIZUKA 1988;
[11] Takagi,H., Hayashi,I; *Artificial_Neural_Network - Driven Fuzzy Reasoning*, Int. Workshop on Fuzzy Systems App., IIZUKA 1988;
[12] Widrow,B., Stearns,S.; *Adaptive Signal Processing*, Prentice Hall, 1885;
[13] Yamaguchi,T., Tanabe,M., Kuriyama,K., Mita,T.; *Fuzzy Associative Memory Application to Adaptive Control*, Proc. of IFSA 1991.

N93

29558

UNCLAS

FUZZY LEARNING UNDER AND ABOUT AN UNFAMILIAR FUZZY TEACHER

Belur V. Dasarathy
Dynetics, Inc.
P. O. Drawer 'B'
Huntsville, Al 35814-5050

ABSTRACT

This study addresses the problem of optimal parametric learning in unfamiliar fuzzy environments. Prior studies in the domain of unfamiliar environments, which employed either crisp or fuzzy approaches to model the uncertainty or imperfectness of the learning environment, assumed that the training sample labels provided by the unfamiliar teacher were crisp, even if not perfect. Here, the more realistic problem of fuzzy learning under an unfamiliar teacher who provides only fuzzy (instead of crisp) labels, is tackled by expanding the previously defined fuzzy membership concepts to include an additional component representative of the fuzziness of the teacher. The previously studied scenarios, namely, crisp and fuzzy learning under (crisp) unfamiliar teacher, can be looked upon as special cases of this new methodology. As under the earlier studies, the estimated membership functions can then be deployed during the ensuing classification decision phase to judiciously take into account the imperfectness of the learning environment. The study also offers some insight into the properties of several of these fuzzy membership function estimators by examining their behavior under certain specific scenarios.

1. INTRODUCTION

Probabilistic decision making in imperfectly supervised environments, i.e., scenarios wherein the labels of the given training samples are unreliable, has been extensively studied in the literature over the past two decades [1-7]. Typical of the learning models proposed are: probabilistic teacher [1], imperfect teacher [2, 3], unfamiliar teacher [4] and VEDIC teacher [5]. A couple of fuzzy models [6, 7] have also been proposed recently. The probabilistic teacher approach proposed by Agrawala [1], which represented a start of a whole new line of studies, essentially disregards the given unreliable labels, i.e., treats the imperfect environment as unsupervised and uses a probabilistic labeling scheme to learn the underlying parameters for the design of the classifier. On the other hand, the imperfect teacher model proposed by Shanmugam [3] assumes that a precise knowledge of the level of imperfection (β) in the environment is available *a priori* and uses this information to guide the parameter learning. This improves the quality of learning over the probabilistic model only as long as the underlying assumption is valid, i.e., the level of imperfection assumed is close to the reality. Otherwise, the resultant learning under the imperfect teacher is likely to be worse than under the probabilistic teacher, which, in essence, assumes $\beta = 0.5$ for a two-class problem, or in a more general case, $\beta = 1/m$, where m is the number of known pattern classes in the environment.

The unfamiliar teacher scheme reported by Dasarathy and Lakshminarasimhan [4] avoids both of these complementary problems, of either having to disregard the imperfect labels entirely and lose some useful information or making a possibly wrong assumption on the imperfectness level and thereby biasing the learning process. This is accomplished by viewing the environment initially as unknown (i.e., starts the learning process in much the same manner as the probabilistic teacher scheme, with β as 1/m) and then learning β about the environment simultaneously with the learning of the parameters for the classifier system design. This learning about the teacher has been shown to aid and enhance the learning under the teacher (see Figure 2 of reference [4]). This approach was extended by Dasarathy and Lakshminarasimhan [5] to dynamic scenarios using the VEDIC teacher model, wherein the level of imperfection β, in addition to being unknown, is also changing with time.

Recently, fuzzy models [6, 7] were proposed to effectively capture the uncertainties caused by the imperfectness of these crisp teachers. Underlying these fuzzy techniques is the need to define a fuzzy membership matrix for the given training data set. Various approaches have been proposed for learning these membership functions. The objective of the study being reported here is to adapt this novel concept of fuzzy learning under an unfamiliar teacher to the problem of learning in an environment that is not only unfamiliar, i.e. labeling information is of unknown level of reliability, but also fuzzy, i.e., the given training samples are associated with multiple classes (rather than just one) with membership distributed across the pattern classes. The fuzzy membership functions to be learnt during the training phase reflect not only the inherent imperfectness but also the fuzziness in the class association provided by the unfamiliar fuzzy teacher. These fuzzy memberships can then be used in the classification phase to appropriately bias the decision process. Details of this integration of the concepts of fuzzy learning under an unfamiliar crisp teacher with those of a fuzzy teacher are presented in the sequel. Section 2 briefly reviews the basic crisp learning under an unfamiliar crisp teacher. Section 3 provides a short overview of fuzzy learning under an unfamiliar but still crisp teacher. In section 4, the adaptation and extension of these ideas to the scenario of fuzzy learning under an unfamiliar fuzzy teacher is presented. The associated algorithmic procedure is outlined in section 5 to aid the implementation of the new methodology. Section 6 outlines some potential alternatives to the initially proposed fuzzy membership model. The last section offers some concluding comments.

2. CRISP LEARNING UNDER AN UNFAMILIAR CRISP TEACHER

The intuitively appealing concept, of learning about an unfamiliar teacher as an aid to learning under the teacher, that underlies this study, was first proposed and successfully demonstrated by Dasarathy and Lakshminarasimhan [4] in 1976 in a two-class crisp environment. They showed that this learning under an unfamiliar teacher is indeed an efficient and practical tool for learning in imperfectly supervised environments wherein it is unrealistic to assume that the level of imperfectness is known *a priori*, the basis of earlier studies in this area. This dual learning process, of learning about the teacher concurrently with parametric learning under the teacher, is schematically illustrated in Figure 1.

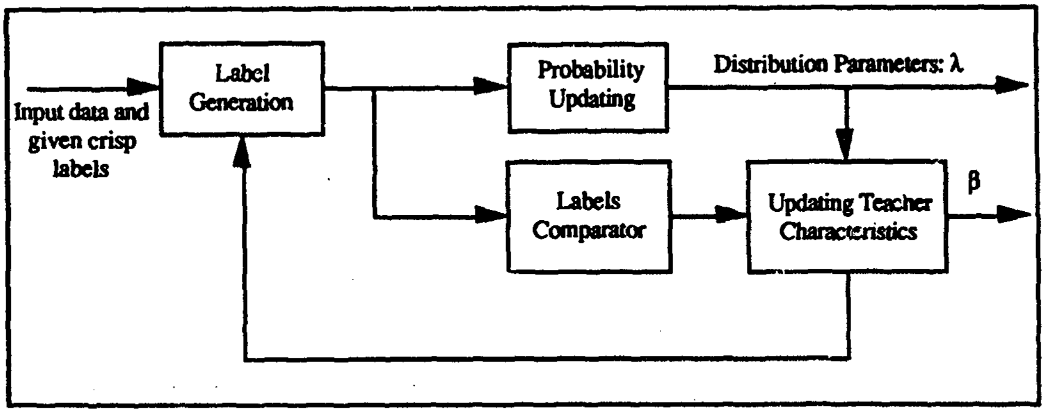

Input data and given crisp labels
Label Generation
Probability Updating
Distribution Parameters: λ
Labels Comparator
Updating Teacher Characteristics
β

Figure 1. Crisp Learning in Unfamiliar Crisp Teacher Environments

Here the learning about the teacher consists in learning β, the effective level of imperfectness in the labels provided by the teacher (environment). This is modeled as a Bernoulli trial with parameter β and a Bayes estimator for minimum quadratic loss, which has a beta distribution [8], is set up for the estimation of β. The learning under the teacher consists in learning the parameters of the underlying distributions which is essential for classification in the operational phase, the primary objective of the effort. It is to be noted that this learning scheme [4] in essence encompasses the spectrum of learning scenarios, starting from learning with a perfect teacher ($\beta = 1$) up to learning without a teacher or learning with a probabilistic teacher ($\beta = 1/m$) through learning with a known imperfect teacher (i.e., β, $1/m \leq \beta \leq 1$ is known *a priori*) and ultimately learning under the most realistic of these scenarios namely learning with an unfamiliar teacher, i.e., β, ($1/m \leq \beta \leq 1$) is unknown *a priori* and is learnt simultaneously with parametric learning. Further implementation details of this learning process can be gleaned from [4] and, as such, are not presented here to save on publication space.

3. FUZZY LEARNING UNDER AN UNFAMILIAR CRISP TEACHER

The unfamiliar teacher scheme discussed in the previous section was synergistically combined recently [6] with the now-well-understood concepts of fuzzy membership to derive a potentially powerful tool of fuzzy learning in unfamiliar teacher environment. This integrated learning is schematically illustrated in Figure 2. Here, the learning includes, not only the distribution parameters and the imperfectness level (as outlined in the previous section), but also the fuzzy membership values generated for each of the input samples by the fuzzy modeling of the uncertainties in the learning environment. This synergism permits the user to exploit the benefits of both the unfamiliar teacher hypothesis as well as those of fuzzy learning concepts. The algorithmic and other details of this integrated scheme of learning, as well as the associated fuzzy membership models, their alternatives and properties, being readily available in the study published recently [6], are not repeated here in the interest of conservation of publication space.

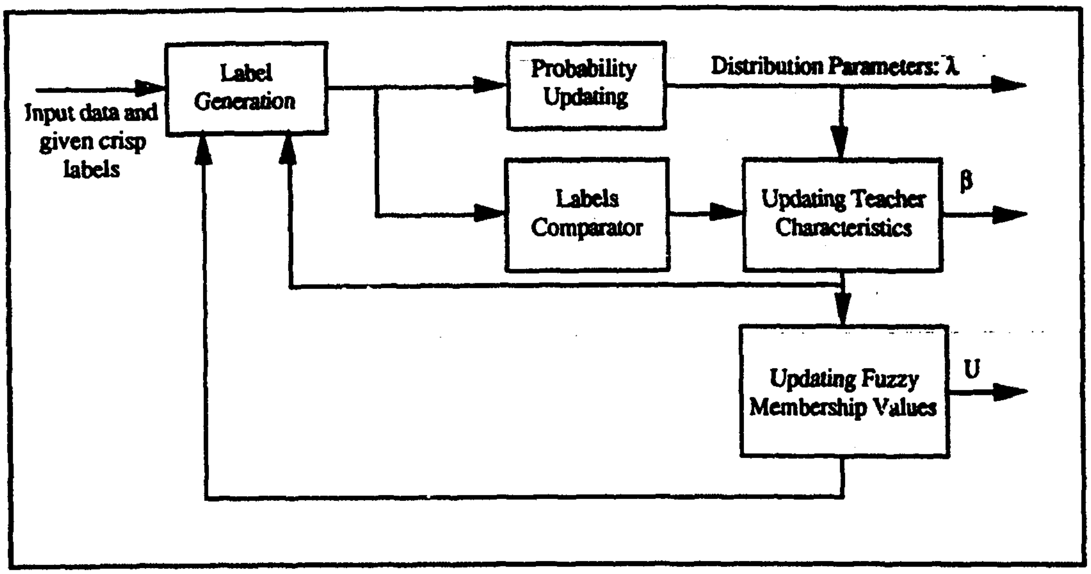

Input data and given crisp labels
Label Generation
Probability Updating
Distribution Parameters: λ
Labels Comparator
Updating Teacher Characteristics
β
Updating Fuzzy Membership Values
U

Figure 2. Fuzzy Learning in Unfamiliar Crisp Teacher Environments

4. FUZZY LEARNING UNDER A UNFAMILIAR FUZZY TEACHER

The scheme of fuzzy learning under an unfamiliar teacher, outlined in the previous section, assumed that the labels provided by this unfamiliar teacher, were crisp, even if imperfect. However, in real-world environment, the imperfect teacher is likely to be fuzzy also. The previously reported fuzzy model [6], which was postulated to take into account only the imperfectness of the teacher, had no provision for taking into consideration the fuzziness in the teacher behavior. Accordingly, a more generalized fuzzy membership model, viewed as the sum of two weighted components is proposed here. This new learning process is schematically shown in Figure 3.

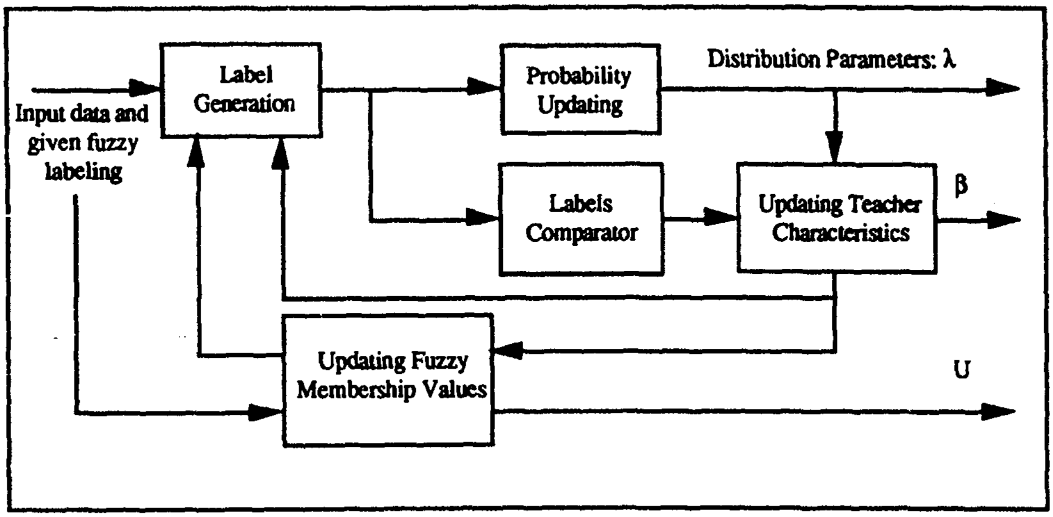

Input data and given fuzzy labeling
Label Generation
Probability Updating
Distribution Parameters: λ
Labels Comparator
Updating Teacher Characteristics
β
Updating Fuzzy Membership Values
U

Figure 3. Fuzzy Learning in Unfamiliar Fuzzy Teacher Environments

The new fuzzy membership function, in effect, captures both the imperfectness and fuzziness of the unfamiliar teacher environment during the learning phase. This is then used to correspondingly weight the decisions made in the classification phase. This is the central idea of the methodology presented here in this study. The recursive learning process necessary for accomplishing this objective can be viewed as one of upgrading the fuzzy membership values furnished by the teacher for each training sample, simultaneously with the learning of the underlying distribution parameters and the level of imperfectness of the supervision available in the environment.

The input to this recursive triple learning process consists of :

- a set of training samples or feature vectors $\{x_i : i = 1, \ldots n\}$
- a set of corresponding fuzzy label memberships $\{\{ v_{ij}; i = 1, \ldots n \}; j = 1, \ldots, m\}$

These labels memberships are assumed to have a level of reliability β which is unknown at the start of the learning process and is learnt during the learning process simultaneously with the parameters of the underlying distributions. This learning begins with an assumption of $\beta = 1/m$, i.e., the labels are essentially disregarded. Thus, initially each sample would have membership values in all the given classes in proportion to the *a priori* probabilities of these classes in the environment since we do not as yet have any measure of confidence in the unfamiliar teacher furnished fuzzy labeling information. Under equal *a priori* probabilities of the classes, the membership function values for each sample will be 1/m provided, of course, the environment is completely exposed, i.e., all the classes expected in the environment are represented in the training set. Otherwise, one will have to adapt into this scenario additional concepts such as learning in partially exposed environments [9] that have been developed for dealing with cases wherein all the classes are not representatively known at the start of the learning process. This would involve adding the flexibility of a reject option to the classification phase and hence a method of defining or learning the boundaries of the currently known classes relative to the rest of the world in addition to learning the boundaries between the known classes. While this is conceivable in the light of the reported developments [9], it is not considered here as being outside the scope of the current study.

As the recursive learning progresses, each sample is assigned probabilities of belonging to the different classes by the unfamiliar teacher scheme (in a manner similar to equation (4) of reference [4] but modified to take into account multiple classes and the current fuzzy membership function values to correspondingly weight the different *a priori* probabilities). Then, one can update class fuzzy membership values based on not only the teacher furnished fuzzy membership values, but also on the relative proportionalities of the *a posteriori* probabilities and β the imperfectness of the unfamiliar teacher. Let p_{ij} be the *a posteriori* probability of x_i being assigned to class j computed on the basis of not only the feature vector values but also the current fuzzy membership functions and teacher imperfectness measure. Then the updated membership function value u_{ij}, of x_i being in class j, is given in terms of the two weighted components as shown in expression (1):

$$u_{ij} = \alpha\, v_{ij} + (1 - \alpha)\, w_{ij} \qquad (1)$$

where

$$w_{ij} = f(\beta, m, p_{ij}, j = 1, \ldots, m) = \begin{cases} \dfrac{\beta\, p_{ij}}{\left[\beta\, p_{iL_i} + \dfrac{(1-\beta)}{(m-1)} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} p_{ik}\right]}; & j = L_i \\[3em] \dfrac{\dfrac{(1-\beta)}{(m-1)}\, p_{ij}}{\left[\beta\, p_{iL_i} + \dfrac{(1-\beta)}{(m-1)} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} p_{ik}\right]}; & j = 1, \ldots, m \\ & j \neq L_i \end{cases} \qquad (2)$$

Here, the first component, v_{ij} is the one furnished by the fuzzy teacher. The second component, w_{ij}, is determined by the learning system (in a manner similar to that proposed in the previous study) to account for the imperfectness aspect of the unfamiliar teacher and α is the relative weighting of the two components. When $\alpha = 0$, this effectively corresponds to the scenario studied previously in [6] with the unfamiliar teacher providing crisp labels. At the other end of the spectrum, i.e. $\alpha = 1$, we only have the fuzziness defined by the teacher with no term to take into account the imperfectness of the teacher within the fuzzy membership model (crisp learning under a fuzzy teacher ! - a not very convincing model of learning). A conceptually elegant choice for α is given by the equation (3):

$$\alpha = \frac{(m\beta - 1)}{(m - 1)} \tag{3}$$

Here, as $\beta \Rightarrow 1$, i.e. as the teacher progressively becomes more and more reliable, the imperfectness in the labeling reduces, $\alpha \Rightarrow 1$, more reliance is placed on the teacher provided fuzzy label information (v_{ij}) and less on the recursively determined component (w_{ij}). On the other hand, as $\beta \Rightarrow 1/m$, i.e. the teacher becomes less reliable and tends towards the unsupervised scenario, $\alpha \Rightarrow 0$, the fuzzy membership information provided by the teacher becomes less relevant and more weight is given to the component determined by the actual a posteriori probabilities.

Equation (1) can be rewritten using equation (3) as

$$u_{ij} = \frac{(m\beta - 1)}{(m - 1)} v_{ij} + \frac{m(1 - \beta)}{(m - 1)} w_{ij} \tag{4}$$

Here, we have

$$\sum_{j=1}^{m} p_{ij} = 1 \tag{5}$$

Substituting equation (5) in the expression (2) we can rewrite (2) as

$$w_{ij} = \begin{cases} \dfrac{(m - 1)\beta p_{ij}}{\left[(1 - \beta) + (m\beta - 1) p_{iL_i}\right]} & ; \ j = L_i \\[2ex] \dfrac{(1 - \beta) p_{ij}}{\left[(1 - \beta) + (m\beta - 1) p_{iL_i}\right]} & ; \ \begin{array}{l} j = 1, \dots, m \\ j \neq L_i \end{array} \end{cases} \tag{6}$$

For the special case of m = 2, which corresponds to the classical detection or binary decision problem, expression (6) reduces to

$$w_{ij} = \begin{cases} \dfrac{\beta p_{iL_i}}{\left[(1 - \beta) + (2\beta - 1) p_{iL_i}\right]} & ; \ j = L_i \\[2ex] \dfrac{(1 - \beta)(1 - p_{iL_i})}{\left[(1 - \beta) + (2\beta - 1) p_{iL_i}\right]} & ; \ j \neq L_i \end{cases} \tag{7}$$

Here, it is interesting to note that in equation (6), $w_{ij} : j = L_i$ is symmetrically dependent on β and $p_{ij} : j = L_i$. As the supervision improves, i.e., as β increases towards unity, the fuzziness due to imperfectness reduces (the membership function component w_{ij} approaches unity for the class corresponding to the given label), but its relative weightage in equation (4) reduces. When the a posteriori probability increases (for a given imperfectness level of the supervision), the component $w_{ij} : j = L_i$, once again approaches unity, and thereby contributes to a corresponding increase in $u_{ij} : j = L_i$ also, since β is not decreasing. Thus, although this second component ($w_{ij} : j = L_i$) is symmetric with respect to the imperfectness level and a posteriori probability, the total fuzzy membership function ($u_{ij} : j = L_i$) is not symmetric.

At the other end of the spectrum, when $\beta = 1/m$, i.e., with essentially no supervision, expression (4) reduces to:

$$u_{ij} = w_{ij} = p_{ij} \quad ; \forall j = 1, \ldots m \tag{8}$$

Thus, under the unsupervised scenario, the fuzzy membership values are dictated wholly by the relative *a posteriori* probabilities of the sample belonging to different classes computed on the basis of the estimated values of the distribution parameters. For all other values of β in the range $(1/m) < \beta < 1.0$, the fuzzy membership value is a function of both the relative *a posteriori* probabilities as well as the reliability level of the labels provided by the teacher as given by equations (4) and (6). Since p_{ij} and β can be computed during the sequential learning scheme based on the unfamiliar teacher concepts (using appropriately modified forms of equations (4) and (5) in Reference [4]), we can continually update w_{ij} and hence u_{ij} also. This construct also assures consistency with the definition of these functions, i.e., the sum of membership function values for every sample is equal to unity. For cases wherein the *a posteriori* probabilities p_{ij} are all equal for a given sample x_i, (i.e., $p_{ij} = 1/m$ for all $j = 1, \ldots, m$) expression (6) reduces to:

$$w_{ij} = \begin{cases} \beta & ; \ j = L_i \\ \dfrac{(1 - \beta)}{(m - 1)} & ; \ j \neq L_i \end{cases} \tag{9}$$

We can also derive the case for which the second component of all the membership functions become equal, i.e.,

$$w_{ij} = w_i \ \forall j = 1, \ldots, m \tag{10}$$

as

$$p_{ij} = \frac{(m - 1)\,\beta\, p_{iL_i}}{(1 - \beta)} \quad ; \quad \beta \neq 1 \tag{11}$$

Here, it is interesting to note that equation (11) reduces to the previously discussed case of equal *a posteriori* probabilities for all classes when $\beta = 1/m$.

In view of fact that the sum of the *a posteriori* probabilities of all classes is unity (equation (5)), equation (11) in effect defines a specific value for the *a posteriori* probability as:

$$p_{iL_i} = \frac{(1 - \beta)}{[1 + m\,(m - 2)\,\beta]} \tag{12}$$

Correspondingly, equation (11) becomes

$$p_{ij} = \frac{(m - 1)\,\beta}{[1 + m\,(m - 2)\,\beta]} \tag{13}$$

Equation (12) reduces to $1/m$ for $\beta = 1/m$ and $(1 - \beta)$ for $m = 2$ the two special cases previously considered here. Correspondingly, equation (13) reduces to $1/m$ for $\beta = 1/m$ and β for $m = 2$.

The outputs of this sequential fuzzy learning (under the unfamiliar fuzzy teacher) process are:

- λ - the parameters of the underlying distributions of the different pattern classes in the environment
- β - the level of imperfectness of the unfamiliar fuzzy teacher in the environment
- U - the set of membership function values for all the training samples over all the different classes

5. ALGORITHMIC PROCEDURE

(i) Set $\beta = \beta_0$ at an appropriate level ($1/m \leq \beta \leq 1$) - the scheme is not very sensitive to this initial value and as such the choice is not critical

(ii) As each new sample x_i is input, generate a probabilistically assigned label ℓ_i based on the parameter estimates as known using the fuzzy membership function values to appropriately weight the corresponding classes (by modifying the multiclass version of equation (4) of reference [4])

(iii) Update the parameters λ (using equation (3) of reference [4])

(iv) If the teacher furnished label L_i and the generated label ℓ_i match, set $Y_i = 1$ denoting a success event; else set $Y_i = 0$

(v) Update β (using equation (5) of reference [4])

(vi) Update w_{ij} using equation (6)

(vii) Update u_{ij} using equation (4)

(viii) Go back to step (ii) for the next sample

(ix) Repeat the procedure till are the samples have been processed

6. SCOPE FOR EXTENSIONS

In the analysis hitherto, it was assumed that the imperfectness β of the unfamiliar teacher is constant across the different pattern classes. However, in the real-world environment, this may not always be true since information acquired about some of the pattern classes may be less reliable than that available for other classes. For example, in the field of non-cooperative target recognition, it is quite likely that the labels of samples acquired of friendly targets may be more reliable than those of adversaries. These differences in the environment have to be appropriately taken into account by the learning process for the learning to be truly optimal. Accordingly, the recursive learning process will have to be suitably modified to accommodate different β values for the different classes. Here, these multiple β's are assumed to be statistically independent for the limited purpose of the ensuing discussions. However, appropriate models which accommodate statistical dependence between these β's can also be visualized, if so desired. Under this class dependent β scenario, multiple estimators (similar to the equation (5) in reference [4]) will have to be set up, one for each of the pattern classes expected in the environment. Correspondingly, expression (2), employed here for estimating the fuzzy membership functions, will also have to be modified to permit multiple β values. This is accomplished by employing an estimator illustrated by the following expression:

$$w_{ij} = f(\beta_j, m, p_{ij}, j = 1, \ldots, m) = \begin{cases} \dfrac{\beta_{L_i} p_{ij}}{\left[\beta_{L_i} p_{iL_i} + \dfrac{(1-\beta_{L_i})}{(m-1)} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} p_{ik} \right]} & ; \; j = L_i \\[4ex] \dfrac{\dfrac{(1-\beta_{L_i})}{(m-1)} p_{ij}}{\left[\beta_{L_i} p_{iL_i} + \dfrac{(1-\beta_{L_i})}{(m-1)} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} p_{ik} \right]} & ; \; \begin{array}{l} j = 1, \ldots, m \\ j \neq L_i \end{array} \end{cases} \tag{14}$$

However, in order to ensure that the summation across the classes of the membership function u_{ij} equals unity in general for each sample, the weights of the two components in equation (1) have to be constant and independent of the class. Hence, equation (3), as defined earlier, can not be validly employed whenever β is not constant across the pattern classes. Accordingly, equation (3) is modified to be:

$$\alpha = \frac{(\sum_{j=1}^{m} \beta_j - 1)}{(m - 1)} = \frac{(m <\beta> - 1)}{(m - 1)} \qquad (15)$$

where <β> is the average value of β across the classes.

Using equation (5), expression (14) can again be restructured as

$$w_{ij} = \begin{cases} \dfrac{(m - 1)\beta_{L_i}\, p_{ij}}{\left[(1- \beta_{L_i}) + (m\, \beta_{L_i} - 1)\ p_{iL_i}\right]} & ;\ j = L_i \\[2ex] \dfrac{(1- \beta_{L_i})\, p_{ij}}{\left[(1- \beta_{L_i}) + (m\, \beta_{L_i} - 1)\ p_{iL_i}\right]} & ;\ \begin{matrix} j = 1, \ldots, m \\ j \neq L_i \end{matrix} \end{cases} \qquad (16)$$

As before, for the case m = 2, i.e., a binary decision case with different levels of reliability for the labels of the target samples from the two classes (for example lethal and benign), we have

$$w_{ij} = \begin{cases} \dfrac{\beta_{L_i}\, p_{ij}}{\left[(1- \beta_{L_i}) + (2\, \beta_{L_i} - 1)\, p_{iL_i}\right]} & ;\ j = L_i \\[2ex] \dfrac{(1- \beta_{L_i})\, p_{ij}}{\left[(1- \beta_{L_i}) + (2\, \beta_{L_i} - 1)\, p_{iL_i}\right]} & ;\ j \neq L_i \end{cases} \qquad (17)$$

Expression (9), which represents the case of equal *a posteriori* probabilities p_{ij}, will also get modified correspondingly, whenever the imperfectness levels for the different classes are estimated separately. Expression (14) reduces to:

$$w_{ij} = \begin{cases} \beta_{L_i} & ;\ j = L_i \\[1ex] \dfrac{(1- \beta_{L_i})}{(m - 1)} & ;\ j \neq L_i \end{cases} \qquad (18)$$

Again, for the second component of all the different class memberships of a given sample to be equal, (equation (10)) can be derived from expression (14) as

$$p_{ij} = \frac{(m - 1)\, \beta_{L_i}\, p_{iL_i}}{(1- \beta_{L_i})} \ \forall\ j \ \begin{matrix} = 1, \ldots, m \\ \neq L_i \end{matrix} \quad ;\ \beta_{L_i} \neq 1 \qquad (19)$$

which as before is again subject to the constraint equation (5). Hence, from equations (5) and (19), we get

$$p_{iL_i} = \frac{(1 - \beta_{L_i})}{\left[m\,(m - 2)\, \beta_{L_i} + 1\right]} \ ;\ \beta_{L_i} \neq 1 \qquad (20)$$

For the binary decision case, i.e., m = 2, this reduces to

$$\begin{aligned} p_{iL_i} &= (1 - \beta_{L_i})\,; \\ p_{ij} &= \beta_{L_i} \qquad ;\ j \neq L_i \end{aligned} \tag{21}$$

The corresponding steps of the algorithmic procedure outlined in section 5 should be appropriately modified to reflect this variable nature of β across the classes, assuming the β's to be statistically independent.

In expression (14), the fuzzy membership component w_{ij} is a function of only the imperfectness of the class represented by the given label, i.e., it is independent of the quality of supervision available for classes other than to which the sample is assigned by the teacher. A more realistic, but complex, model would be of the form:

$$w_{ij} = \begin{cases} \dfrac{\beta_{L_i} p_{ij}}{\left[\beta_{L_i} p_{iL_i} + \dfrac{m(1-\beta_{L_i})}{(m-1)^2} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} (1-\beta_k) p_{ik} \right]} & ;\ j = L_i \\[4ex] \dfrac{\dfrac{m(1-\beta_{L_i})}{(m-1)^2} (1-\beta_j) p_{ij}}{\left[\beta_{L_i} p_{iL_i} + \dfrac{m(1-\beta_{L_i})}{(m-1)^2} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} (1-\beta_k) p_{ik} \right]} & ;\ \begin{matrix} j = 1, \ldots, m \\ j \neq L_i \end{matrix} \end{cases} \tag{22}$$

It is interesting to note that in expression (22), w_{ij} becomes a function of the imperfectness levels of all the different classes in the environment while retaining the uniqueness of the original expression (14) for the perfectly supervised class case of $\beta_{L_i} = 1$. (However, unlike equations (2) and (14), equation (22) cannot be restructured to eliminate the summation over p_{ik} because of the presence of the variable β_k within the summation term). Expression (22) can therefore be viewed as a more realistic portrayal of the imperfectness in the environment for the classification phase. Under this model, a crisp (i.e. 1 or 0 value) scenario gets established for w_{ij} whenever just the single corresponding imperfectness parameter disappears ($\beta_j = 1$). This therefore represents an .OR. logic based dependence across the imperfectness values. One can also visualize an .AND. logic based version of the model as:

$$w_{ij} = \begin{cases} \dfrac{\beta_{L_i} p_{ij}}{\left[\beta_{L_i} p_{iL_i} + \dfrac{1}{2(m-1)} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} (2-\beta_{L_i}-\beta_k) p_{ik} \right]} & ;\ j = L_i \\[4ex] \dfrac{\dfrac{(2-\beta_{L_i}-\beta_j)}{2(m-1)} p_{ij}}{\left[\beta_{L_i} p_{iL_i} + \dfrac{1}{2(m-1)} \displaystyle\sum_{\substack{k=1 \\ \neq L_i}}^{m} (2-\beta_{L_i}-\beta_k) p_{ik} \right]} & ;\ \begin{matrix} j = 1, \ldots, m \\ j \neq L_i \end{matrix} \end{cases} \tag{23}$$

7. CONCLUDING COMMENTS

The study offers a potent tool for learning and operating in imperfectly supervised fuzzy environments. This is accomplished by treating the fuzzy environment as essentially unfamiliar at the initiation of the learning process and thereafter learning about the environment in terms of the level of imperfectness and the fuzzy membership values for each training sample concurrently with the primary learning task of determining the underlying distribution parameters. The major innovation of this study is the development of an unique concept of jointly capturing within the defined fuzzy framework both the imperfectness of the unfamiliar teacher as well as the fuzziness in the labeling provided. Admittedly, alternative fuzzy model formulations (such as for example equation (24)), can easily be conceived.

$$w_{ij} = \begin{array}{ll} \dfrac{\prod_{k=1}^{m} \beta_k \, p_{ij}}{\left[\prod_{k=1}^{m} \beta_k \, p_{iL_i} + \dfrac{1}{2(m-1)} \sum_{\substack{k=1 \\ \neq L_i}}^{m} (2 - \beta_{L_i} - \beta_k) \, p_{ik} \right]} & ; \; j = L_i \\[4ex] \dfrac{\dfrac{(2 - \beta_{L_i} - \beta_j)}{2(m-1)} \, p_{ij}}{\left[\prod_{k=1}^{m} \beta_k \, p_{iL_i} + \dfrac{1}{2(m-1)} \sum_{\substack{k=1 \\ \neq L_i}}^{m} (2 - \beta_{L_i} - \beta_k) \, p_{ik} \right]} & ; \; \begin{array}{l} j = 1, \ldots, m \\ j \neq L_i \end{array} \end{array} \qquad (24)$$

The approach can also be extended to dynamic environments by combining the VEDIC teacher concepts [5,7] with the dual-component based membership function learning methodology developed here.

ACKNOWLEDGMENTS

The author would like to take this opportunity to gratefully acknowledge the IR&D funding support provided for this study by Mr. Tom Baumbach, Executive Vice-President, Dynetics, Inc.

REFERENCES

[1] A. K. Agrawala, "Learning with a Probabilistic Teacher," *IEEE Transactions on Information Theory*, Vol. IT-16, No. 4, pp. 373-379, July 1970.

[2] K. Shanmugam and A. M. Breiphol, "An Error Correcting Procedure for Learning with an Imperfect Teacher," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-1, No. 3, pp. 223-229, July 1971.

[3] K. Shanmugam, "A Parametric Procedure for Learning with an Imperfect Teacher," *IEEE Transactions on Information Theory*, Vol. IT-18, No. 2, pp. 300-302, March 1972.

[4] B. V. Dasarathy and A. L. Lakshminarasimhan, "Sequential Learning Employing Unfamiliar Teacher Hypotheses (SLEUTH) With Concurrent Estimation of Both the Parameters and Teacher Characteristics," *International Journal of Computers and Information Sciences*, Vol. 5, No. 1, pp. 1-7, March 1976.

[5] B. V. Dasarathy and A. L. Lakshminarasimhan, "Learning Under a VEDIC Teacher," *International Journal of Computer and Information Sciences*, Vol. 8, No. 1, pp. 75-88, March 1979.

[6] B. V. Dasarathy, "FLUTE: Fuzzy Learning in Unfamiliar Teacher Environments," *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, CA., pp. 1070-1077, March 1992.

[7] B. V. Dasarathy, "Fuzzy Learning in Vicissitudinous Environments," *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, The Hague, Netherlands, August 1992.

[8] E. A. Patrick, *Fundamentals of Pattern Recognition*, Prentice Hall, 1972.

[9] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, IEEE Computer Press, California, Chapter 5, pp. 235-284, 1991.

N93

29559

UNCLAS

Some Problems with the Design of Self-Learning Management Systems

Ziny Flikop
NYNEX Science and Technology, Inc.
500 Westchester Avenue, White Plains, NY, 10604 USA

Abstract

In this paper some problems in the design of management systems for complex objects are discussed. Considering the absence of adequate models and the fact that human expertise in the management of non-stationary objects becomes obsolete quickly, the use of self-learning together with a two-step optimization of on-line control rules is suggested. To prepare for the object analysis, a set of definitions has been proposed. Traditional and fuzzy sets [1, 2] approaches are used in the analysis. To decrease the reaction time of the control system, we propose the development of control rules without feedback.

Keywords: Control Processes, Decision Theory, Fuzzy Sets, Optimization

1 Introduction

Automatic and semi-automatic control and management systems usually are based on sets of control rules. The development of such rules requires either comprehensive human expertise or an adequate object model or both. However, the design of reliable models of complex objects is often a very difficult task and human expertise in the control of the non-stationary objects becomes obsolete with time. The traditional use of a control with feedback results in prolonged reaction time. These problems can be partially avoided if a self-learning approach is used for the development of control rules. In this paper, which is but another drop in the sea of control and management literature, we too are proposing and studying some variant of such an approach.

The methodology for synthesizing management systems depends on the complexity of the controlled object. In this paper we will discuss the management of complex controlled objects with multiple inputs and outputs. (Queueing networks and assembly plants are examples of such objects.) The object transformation function (mapping) is defined by an object organization (structure) and by the values of the object element set-up parameters (values of control variables). Object performance is evaluated by multiple criteria (via multiple controlled variables). Management system performance is evaluated by the management system's ability to maintain outputs at the predefined level (the simple control task) and by its ability to minimize the "cost" required for the control (the optimized control task). Modern management systems are based on relatively powerful computers and execute their tasks by varying either control variables and/or object structure. In the development of management systems, one should consider that management system reaction time must be much shorter than input drift, and processes of environmental and structural changes.

The ability of management systems to control and optimize objects depends on the efficiency of the algorithms used for these purposes. In turn, these algorithms depend on continuity, separability, and monotonicity of controlled object mapping functions. In general, the behavior of mapping functions depends on the nature of controlled objects. If, for example, simple physical devices often have mapping functions that are continuous, separable, and monotonic, this is not always the case for more complex controlled objects. When mapping functions are continuous, separable, and monotonic, relatively simple control and optimization algorithms can be applied. However, when mapping functions are not continuous, separable, and monotonic, then significantly more powerful algorithms are needed. Because the behavior of mapping functions imposes limitations on the selection of control and optimization algorithms, and since proposed self-learning methodology is based on such algorithms, let's define what we mean by continuity, separability, and monotonicity. Let's also define several other terms used in this paper.

2 Definitions

From a management point-of-view, an object can be defined via the following mapping functions:

$$F(V,H): X \to Y,\ F(X,H): V \to Y \text{ and } F(X,H): V \to Q \tag{1}$$

where:

$X = (x^1, x^2, ..., x^m)$ is an input vector $m = |M|$, M is a set of input variables;
$Y = (y^1, y^2, ..., y^n)$ is an output vector $n = |N|$, N is a set of output variables;
$V = (v^1, v^2, ..., v^p)$ is a vector of control variables, $p = |P|$, P is a set of control variables;
$Q = (q^1, q^2, ..., q^b)$ is a vector of controlled variables, $b = |B|$, B is a set of controlled variables;
$H = \{h_l\}$ is a controlled object structure, h_l is an object element, $l \in L$.

We can define some properties of these functions, which are useful for management system development:

a. Let's consider a function $F(V, H): (X+\lambda) \to (Y+\mu)$ with any fixed V and H as continuous into space C if when $\lambda \to 0$, then $\mu \to 0$ for $\forall X \in C$, $(X+\lambda) \in C$. (2)

b. Let's consider a function $F(X, H): (V+\eta) \to (Y+\mu)$ with any fixed X and H as continuous into space A if when $\eta \to 0$, then $\mu \to 0$ for $\forall V \in A$, $(V+\eta) \in A$. (3)

c. Let's consider a function $F(X, H): (V+\eta) \to (Q+\rho)$ with any fixed X and H as continuous into space L if when $\eta \to 0$, then $\rho \to 0$ for $\forall V \in L$, $(V+\eta) \in L$. (4)

d. Let's consider a function $F(V, H): X \to Y$ with any fixed V and H, and $X \in C$ as monotone if when X is changing in one direction along same monotone trajectory into C, then Y is also changing in one direction along a monotonic trajectory into output space. (5)

e. Let's consider a function $F(X, H): V \to Y$ with any fixed X and H and $V \in A$ as monotone if when V is changing in one direction along same monotone trajectory into A, then Y is also changing in one direction along a monotonic trajectory into output space. (6)

f. Let's consider a function $F(X, H): V \to Q$ with any fixed X and H and $V \in L$ as monotone if when V is changing in one direction along same monotone trajectory into L, then Q is also changing in one direction along monotone trajectory into controlled variable space. (7)

g. Let's consider a function $F(V, H): X \to Y$ as separable if $F: (x^1, x^2, ..., x^i + \Delta^i, ..., x^m) \to (Y + \kappa^i)$ and $F: (x^1, x^2, ..., x^j + \Delta^j, ..., x^m) \to (Y + \kappa^j)$ then $F: (x^1, x^2, ..., x^i + \Delta^i, ..., x^j + \Delta^j, ..., x^m) \to (Y + \kappa^i + \kappa^j)$ (8)

h. Let's consider a function $F(X, H): V \to Y$ as separable if $F: (v^1, v^2, ..., v^i + \Delta^i, ..., v^p) \to (Y + \kappa^i)$ and $F: (v^1, v^2, ..., v^j + \Delta^j, ..., v^p) \to (Y + \kappa^j)$ then $F: (v^1, v^2, ..., v^i + \Delta^i, ..., v^j + \Delta^j, ..., v^p) \to (Y + \kappa^i + \kappa^j)$ (9)

i. Let's consider a function $F(X, H): V \to Q$ as separable if $F: (v^1, v^2, ..., v^i + \Delta^i, ..., v^p) \to (Q + \varphi^i)$ and $F: (v^1, v^2, ..., v^j + \Delta^j, ..., v^p) \to (Q + \varphi^j)$ then $F: (v^1, v^2, ..., v^i + \Delta^i, ..., v^j + \Delta^j, ..., v^p) \to (Q + \varphi^i + \varphi^j)$ (10)

We can define the fluctuation range of i, $(i \in M)$ input variable by an ordered A^i set that consists of real numbers s^j representing possible measured values of this variable. The whole input space is:

$$A = A^1 \times A^2 \times ... \times A^m \tag{11}$$

We can define the reference (desired) value y_r^i for every output variable i, $(i \in N)$ and the reference output vector $Y_r = (y_r^1, y_r^2, ..., y_r^n)$ for the whole object.

We also can define the permissible output space:

$$\Phi = [y_r^1 + \alpha^1, y_r^1 - \alpha^1] \times [y_r^2 + \alpha^2, y_r^2 - \alpha^2] \times ... \times [y_r^n + \alpha^n, y_r^n - \alpha^n] \tag{12}$$

where α^i is an accuracy of tracking i variable; $[y_r^i + \alpha^i, y_r^i - \alpha^i]$ is a permissible interval of i output variable.

Each permissible interval also can be represented by a normalized fuzzy set U^i with a membership function $b_{U^i}(y), y \in suppU^i$. In this set, $y_t^i = y_r^i$ has a maximal possible grade $b_{U^i}(y_t^i) = 1$. In this case, the permissible output space Θ can be defined as:

$$\Theta = suppU^1 \times suppU^2 \times \ldots \times suppU^n \qquad (13)$$

The actual output vector Y_t at moment t usually differs from Y_r. This difference is the result of either X_t drift or mapping function changes caused by V instability and environmental and H changes. Deviation of Y_t from Y_r is a control error for which the management system must compensate. Compensation can be done either by varying only V or only H or by simultaneous changes V and H.

The quality of control is evaluated either by an output error vector that at moment t is:

$$\aleph_t = (\varepsilon_t^1, \varepsilon_t^2, \ldots, \varepsilon_t^n), \text{ where } \varepsilon_t^i = y_t^i - y_r^i, \ (i \in N). \qquad (14)$$

or by $\Im = \sum_{i \in N} (1 - b_{U^i}(y_t^i))$ that can be used for estimating a degree that $Y_t \notin \Theta$. (15)

We will consider that the controlled object is working within a required accuracy if either of the following conditions are satisfied:

$$\text{for } \forall i, (i \in N) \ (y_r^i + \alpha^i) < y_t^i > (y_r^i - \alpha^i) \text{ or } y_t^i \in suppU^i \qquad (16)$$

$$\text{or } \ Y_t \in \Phi \ \text{ or } Y_t \in \Theta \qquad (17)$$

All X, for which conditions (16, 17) are satisfied for some combination of VH, are permissible input vectors for this combination. We can propose two definitions of permissible input subspaces Ξ^z for z-th combination of VH:

$$\Xi^z = \{ X | X \rightarrow Y \in \Phi or Y \in \Theta \text{ for } z\text{-th combination of } VH \}$$

$$\text{or } \Xi^z = [x_{max}^{1z} - x_{min}^{1z}] \text{ x } [x_{max}^{2z} - x_{min}^{2z}] \text{ x},\ldots,\text{x}[x_{max}^{mz} - x_{min}^{mz}] \qquad (18)$$

where $x_{max}^{iz} - x_{min}^{iz}$ is a permissible input interval of i input variable which provides conditions (16, 17) for z-th combination;

$x_{max}^{iz}, x_{min}^{iz}$ are correspondingly maximal and minimal permissible values of i input variable.

We also can define each permissible i input interval as a normalized fuzzy set W^{iz} with a membership function $b_{W^{iz}}(x), x \in suppW^{iz}$. Corresponding permissible input space Ψ^z is:

$$\Psi^z = suppW^{1z} \times suppW^{2z} \times \ldots \times suppW^{mz} \qquad (19)$$

We can define for each output variable i at any time t a distance of y_t^i from the border of a permissible output interval either via:

$$\delta_t^i = min\,(|y_t^i + \alpha^i - y_r^i|, |y_r^i + \alpha^i - y_t^i|) \text{ or via membership grades as } 1 - b_{U^i}(y_t^i) \qquad (20)$$

For the output vector we can use either:

$$D_t = \sqrt{\sum_{i \in N} (\delta_t^i)^2} \text{ or (15).} \qquad (21)$$

The efficiency with which the management system executes its controlling functions can be different for different combinations of V and H. We can introduce a multivalued goal function G that can be used for evaluating the efficiency of the management system and optimizing the object:

$$G_t = \sum_{f \in B} \gamma^f q_t^f \qquad (22)$$

where γ^f is a weight coefficient of f controlled variable, $f \in B$.

3 Management

Management systems should be able to work in two interrelated modes:

1. Simple control mode. This is either a process of minimizing an output error vector:

   $$\min_{V,H} \aleph_t = (\varepsilon_t^1, \varepsilon_t^2, \dots, \varepsilon_t^n) \text{ or } \min_{V,H} \Im = \sum_{i \in N} (: - b_{ij}(y_t^i)) \tag{23}$$

   or a process of confining Y_t to permissible output error space (12, 13), which is executed by varying either only V or V together with H. For successful control, conditions (2) and (3) must be satisfied. Controlling algorithms can be relatively simple if conditions (5, 6, 8, 9) are also satisfied.

2. Optimized control mode. This is also a process of the object control corresponding with (24). However, here the object performance is optimized by varying eiuner only V, or V and H:

   $$\max_{V,H} G = \sum_{f \in B} \gamma^f q^f \text{ upon satisfaction (16, 17).} \tag{24}$$

Relatively simple algorithms can be used for this optimization if conditions (4, 7, and 10) are satisfied. If conditions (5-10) are not satisfied, then the algorithm proposed in [3] can be recommended. The optimization (23, 24) that is executed by varying V is based, in general, on non-linear programming. During such an optimization, conditions (16, 17) can be preserved relatively easily. However, the optimization that is executed via controlled object structural changes is based on the combinatorial approach. During combinatorial optimization, conditions (16, 17) can be unexpectedly violated, since any changes of H create a significant destabilization effect on the object. To decrease the possibility of violations of (16, 17), the management system should, before changes of H are made, try to drive Y_t into the center of Φ (Θ). This can be done, for example, via thorough V optimization. The occurrence of Y_t near the center of Φ (Θ) is an indication that the object has an excess of stability. As a result, the optimization executed via changes of H becomes possible.

3.1 Models

Different management system modes require the use of different models. Namely, the simple control mode requires the input-control variables-object structure-output mapping model. This model reflects $F(V, H): X \rightarrow Y$ (Figure 1a) and $F(X, H): V \rightarrow Y$ (Figure1b). For the optimized control mode, a control variables-controlled variables-mapping model that reflects $F'(X, H): V \rightarrow Q$ (Figure 1c) should also be used. Both models are created and updated during the self-learning process.

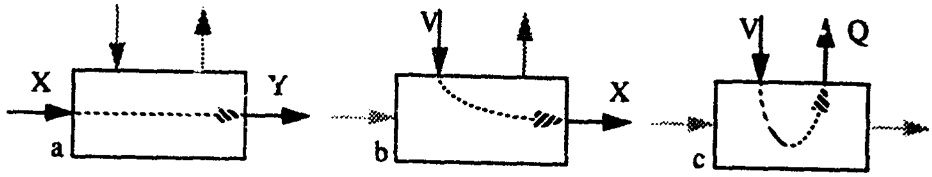


Figure 1

3.2 Control Approaches

For object stabilization, two approaches based either on $V_t H_t = f(Y_t)$ or $V_t H_t = f(X_t)$ can be used. The first approach uses a feedback scheme, i.e., the management system constantly monitors conditions (16, 17) and corrects output, if necessary, by varying V and H. This approach is relatively accurate. However, it is slow, since control decisions are delayed by an object $X \rightarrow Y$ transformation time and by a decision process that requires CPU time. The second $VH = f(X_t)$ approach does not use a feedback scheme all the time. Instead, the management system continuously monitors condition $X_t \in \Xi^z$ ($X_t \in \Psi^z$) and makes decisions, either that the VH combination has to be changed and thus what has to be done to satisfy (16, 17), or that no change has to be made. In other words, the second approach uses rules of this kind: "If $X_t \in \Xi^z$ ($X_t \in \Psi^z$), then do nothing. If $X_t \notin \Xi^z$ ($X_t \notin \Psi^z$), then find other Ξ^r (Ψ^r) to which X_t belongs and change the object in correspondence with $V^r H^r$". This approach is faster than the feedback approach, but requires that $VH = f(X)$ (reactions) for $\forall X$ will be prepared in advance. To decrease reaction time, we will study the possibility of using a second (without feedback) controlling approach in combination with self-learning and adaptation.

4 Self-Learning and Adaptation

The purpose of self-learning is the development (modification) of control rules based on cause-and-effect information received via trials. In the absence of analytical models, it is recommended that trials be made on the real objects. Self-learning consists of three phases. Namely:

1. The preliminary cause-and-effect trials phase. This phase is dedicated to the analysis of how linear $F(V,H)$: $X \to Y$, $F(X,H): V \to Y$, and $F'(X,H): V \to Q$ are, and where conditions (2-7) are satisfied. Conditions (8-10) must also be studied.

   For the study of $F(V,H): X \to Y$ and for the fixed V and H, we will either observe natural X fluctuations on the real object, or actively change X on the model or on the real object. For each X, a value of Y is defined. This process is repeated for different V and H. Similarly, for the study of $F(X,H): V \to Y$, and for the fixed X (if it is possible) and H, we will vary V and define for each V a value of Y. This process is repeated for different X and H. The purpose of this is to check (2, 3, 5, 6, 8, 9).

   For the study of $F'(X,H): V \to Q$, and for the fixed X (if it is possible) and H, we will vary V and define for each V value of Q. This process is repeated for different X and H. The purpose of this is to check (4, 7, 10). The number of such trials is dictated by the desired accuracy of verification of (2-10) and it must be held to the minimum. The results of the first phase are needed for the selection of optimization algorithms.

2. The development of the control rules phase. This is implemented via a two-step object optimization. During the first step, an VH combination (feasible solution) is received for the analyzed X in correspondence with (24) (Figure 2). The second step is the selection of the optimal (in correspondence with (25)) VH combination for the same X. *(The use of optimization for developing rules allows us to restrict the number of analyzed VH combinations only to the combinations used during optimization. As a result, the analysis of all possible combinations of VH, required for the full-scale object analysis, can be avoided.)*

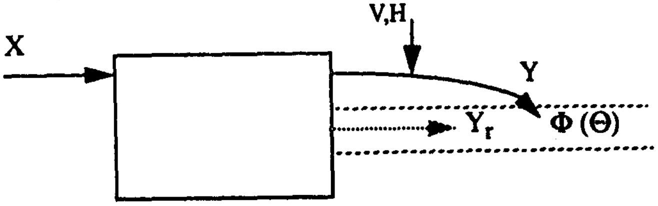


Figure 2

Control rule development starts from some X. When the optimal VH combination is received for this X, then a permissible input space Ξ^z or Ψ^z for this combination is defined on the basis of (18) or (19) by varying either model or real object inputs (Figure 3).

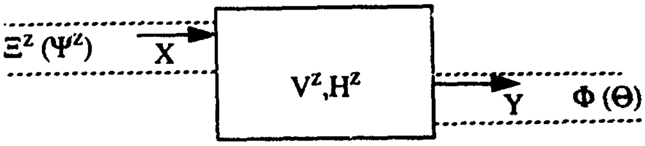


Figure 3

Consequently, we will formulate a rule: "If $X \in \Xi^z$ $(X \in \Psi^z)$, then H^zV^z combination has to be chosen." Then the two-stage optimization process is repeated for some other $X \notin \Xi^z$ $(X \notin \Psi^z)$. As a result, we will find other V^rH^r combinations and other input subspaces Ξ^r or Ψ^r (Figure 4).

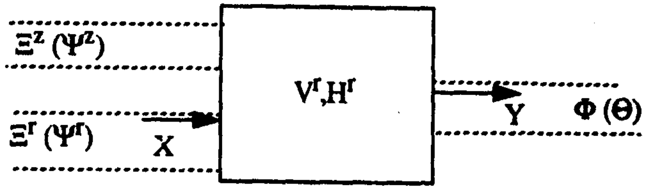


Figure 4

After this, the next X (neither $X \notin \Xi^z$ $(X \notin \Psi^z)$ nor $X \notin \Xi^r$ $(X \notin \Psi^r)$) is selected and the process is repeated. Input vectors are selected for the analysis until either the whole input space A becomes decomposed or a feasible solution for the same X is impossible to find. As a result of decomposition, some input subspaces can intersect; i.e., more than one feasible solution exists for some X; i.e., $X \in \Psi_g, \Psi_g = \bigcap_{z \in Z} \Psi^z_g \neq \emptyset$ or $X \in \Xi_g, \Xi_g = \bigcap_{z \in Z} \Xi^z_g \neq \emptyset$, where Ξ_g or Ψ_g represent a g-th intersection.

If $X_t \notin \Xi^z$, and $X_t \in \Xi_g$ and *if* conditions (3) and (6), or (9) are satisfied, then a combined weighted rule [4] can be executed to provide $Y_t \in \Theta$.

When the controlled object is non-linear, it is possible that $|\Xi^z| \neq |\Xi^r|$ $(|\Psi^z| \neq |\Psi^r|)$. We should try to avoid intersection situations because the more input space that belongs to the intersections, the more VH combinations have to be analyzed. If Ξ^z (Ψ^z) represents an object stability input space for the z-th combination of VH, then $\Xi = \bigcup_{z \in Z} \Xi^z$ $(\Psi = \bigcup_{z \in Z} \Psi^z)$ represents a total object stability input space.

When $\Xi = A$ $(\Psi = A)$, then the object is stable. If $\Xi \in A$ $(\Psi \in A)$, then the object is only partially stable. Results consisting of optimal *input subspace-object structure-control variable vector* rules should be stored in the input-reaction table.

3. The adaptation phase. Because an object and/or object environment usually are non-stationary, the management system's ability to control and its performance efficiency degrade with time. To maintain the controlled object performance at the predefined level, the management system should constantly monitor the validity of the developed rules and the values of G_t. The purpose of monitoring is to detect a moment when management becomes inefficient. When management system inefficiency is detected, another self-learning process is necessary.

Depending on the peculiarities of controlled objects, we will use different approaches in the self-learning. Some controlled objects or their models can be studied in the test-bed mode. In this mode X_t is stabilized as X and precisely measured. Other vectors (Y, V, Q) can be precisely measured also. The test-bed mode permits implementation of a special algorithm during the first and second phases of self-learning. Moreover, self-learning process can be executed automatically. Algorithms for the (23, 24) optimization are selected depending on the results of the cause-and-effect phase. If the test-bed mode is unacceptable, then self-learning has to be implemented on the real object in the on-line mode. In this case conditions (2 and 5) are analyzed within the first phase. This analysis is made via either natural or specially created changes of input variables upon fixed VH. If the algorithm [3] is used for the (23, 24) optimization, then the analysis of conditions (3 and 4) can be made during the second phase. Moreover, algorithm [3] does not require satisfaction of conditions (6 - 10).

References

[1] L. A. Zadeh, "The Concept of a Linguistic Variable and its Application to Approximate Reasoning," *Information Sciences*, 8 pp.199 - 249, 301-357 and 9 pp. 43-80.
[2] M. Sugeno, "An Introductory Survey of Fuzzy Control," *Information Sciences* (1985) 59-83.
[3] Z. Flikop, "Routing Optimization in Packet Switching Communication Networks," *European Journal of Operational Research* 19 (1985) 262-267.
[4] R. R. Yager and D. P. Filev, "Identification of Nonlinear Systems by Fuzzy Models," (forthcoming).

[5] A.S Tanenbaum. Computer Networks, Second edition, Prentice Hall, New Jersey, 1988.

Appendix

This appendix presents an example of the application of the methodology proposed in this paper. Let's consider the packet switching network (Figure 5) analyzed in [5, pp. 304-305]. The network structure H is defined by a set of nodes, their connecting lines, and the line capacities.

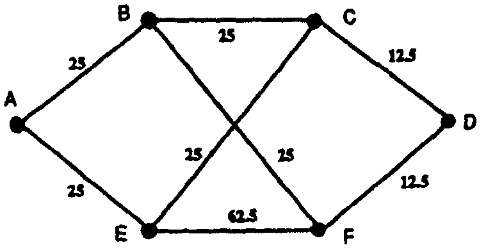


Figure 5

The numbers displayed in Figure 5 represent line capacities C^i (where i is a line number, $i = 1, ..., 16$) in packets-per-second. In this network, traffic transmitted between any source and destination nodes can be split between different paths.

In the described network:

$$F(V_{net}, H) : \Lambda_{inp} \rightarrow \Lambda_{out}$$

where $V_{net} = (v^{AB}, ..., v^{DCBA}, ..., v^{FE})$ is the control vector that is implemented in the network via node routing tables and defines sets of paths for every source-destination pair of nodes. It also defines in what *proportion* traffic must be split between paths. For example, v^{DCBA} is the portion of the traffic transmitted from source node D to destination node A via path $DCBA$. (Control vectors are presented in Table 1, columns "Curve A", "Curve B", and "Curve C".)

$\Lambda_{inp} = (\Lambda^{AB}, \Lambda^{AC}, ..., \Lambda^{BC}, ..., \Lambda^{EA})$ is an input vector, where Λ^{AB}, for example, is traffic that is entering the network via node A and destined for node B. When traffic between nodes A and B is split, then $\Lambda^{AB} = \sum_{j \in J^{AB}} \lambda^j$; J^{AB} is a set of paths between which traffic from A to B is split; λ^j is a traffic in the path j.
One can see examples of splitting in Table 1, "Path" column.

$\Lambda_{out} = (\tilde{\Lambda}^{AB}, \tilde{\Lambda}^{AC}, ..., \tilde{\Lambda}^{BC}, ..., \tilde{\Lambda}^{EA})$ is an output vector. $\tilde{\Lambda}^{AB}$ is, for example, traffic that is entering the network via node A, is transmitted by the network to node B, and is leaving *successfully* the network via node B. (When traffic approaches link capacity, then the network can drop traffic to avoid congestion. In this case $\Lambda_{out} < \Lambda_{inp}$.)

$$F(H, \Lambda_{inp}) : V_{net} \rightarrow \Lambda$$

is a function that defines the mapping of a control vector into a controlled vector
$\Lambda = (\lambda^1, ..., \lambda^i, ..., \lambda^{16})$ that represents the actual (in pkts/sec) traffic in every link of the network.

$$\text{Here } \lambda^i = \sum_{k^i \in K^i} \lambda^{k^i}, \tag{25}$$

where λ^{k^i} is traffic in the k^i path ($k^i \in K^i$); K^i is a set of paths that contain link i. All three parameters (λ^{k^i}, k^i, and K^i) are governed by V_{net}.

$$\text{We evaluate the control } \Phi \text{ of the network via the function: } \Phi = \sum_{i=1}^{16} \lambda^i / (C^i - \lambda^i) \tag{26}$$

For simplicity, let's evaluate an efficiency G of the network according to the following:

$$G = \sum_{i}^{16} C^i \tag{27}$$

For illustration, let's define the network task as one that provides a control upon $\Phi \leq 14.0$ and *minG* when traffic between nodes BF and CE varies and is fixed for all other source-destination pairs of nodes.

When V_{net} provides $C^i > \lambda^i$ for $\forall i$, then mapping $F(V_{net}, H) : \Lambda_{inp} \rightarrow \Lambda_{out}$ is continuous, separable, and monotonic, and $\Lambda_{out} = \Lambda_{inp}$. If a routing optimization algorithm allows for the continuous changing of V_{net}, then mapping $F(\Lambda_{inp}, H) : (V_{net} \rightarrow \Lambda)$ is also continuous. However, it is non-separable because traffic in every link is defined via (25) and (26) is a non-liner function.

For the developing network control rules, we use the model and optimization algorithm proposed in [3]. This model and optimization algorithm allow for the continuous changing of V_{net} and they do not require monotonicity of the optimized function. Development of the rules can be done in the test-bed mode. Since we have already checked the continuity, separability, and monotonicity of the $F(V_{net}, H) : \Lambda_{inp} \rightarrow \Lambda_{out}$ and $F(\Lambda_{inp}, H)$: $V_{net} \rightarrow \Lambda$ mappings, let's go directly to the second phase of the control rules development.

We can start the development with traffic values for $B-F$ and $C-E$ source-destination pairs which are proposed in [6]; namely 4.0 pkts/sec for $B-F$ and 3.0 pkts/sec for $C-E$. During the first step of the optimization (which is based on varying V_{net} upon fixed network structure H), the procedure described in [3] checks the possibility that a control vector, providing conditions $\Phi \leq 14.0$, can be found. Such a control vector "A" was found and is stored in Table 1 in the column, "Curve A traffic (%)". The value of Φ that corresponds to this vector and the analyzed network structure and traffic values is 13.56 and $G = 425$. Since $\Phi < 14.0$, the network has an excess of capacity and its structure can be optimized.

During the second step of optimization, the network structure (link capacities) was changing; however, control vector V_{net} was fixed. As a result of this optimization, the capacity of the EF and FE lines were decreased to 42.63 pkts/sec. That corresponds to $G = 385.26$ and $\Phi = 14.0$. Then permissible input spaces were defined. This was done on a fixed network structure by varying $B-F$ and $C-E$ traffic, while monitoring $\Phi \leq 14.0$ conditions. We started with the original structure and control vector that is optimal for $B-F$ traffic that is equal to 4.0 pkts/sec and $C-E$ traffic that is equal to 3.0 pkts/sec. Modeling allows us to plot curve "A" on Figure 6. The zone under this curve is a permissible input space and it defines possible combinations of $B-F$ and $C-E$ traffic for which the analyzed control vector provides $\Phi \leq 14.0$. In other words, a control rule *"Until $B-F$ and $C-E$ traffic is in the zone below curve "A", the set of routing tables, corresponding to "Curve A traffic (%)" of the Table 1, should be used"* can be applied.

We can see from curve "A" that the maximal $B-F$ traffic is limited to 9.66 pkts/sec and $C-E$ traffic is limited to 8.12 pkts/sec. To analyze the network's ability to absorb more $B-F$ traffic, we can choose a value of $B-F$ traffic which is above 9.66 pkts/sec. Then we can find the other control vector that provides (5) and $\Phi \leq 14.0$ conditions. This vector is presented in Table 1, column "Curve B traffic (%"). Analysis of the network with a new control vector allows us to plot curve "B" in Figure 6.

We can repeat similar procedures for the $C-E$ traffic that exceeds 8.12 pkts/sec. This give us one more control vector (Table 1, column "C") and another curve "C".

As a result of these studies, the following rules can be created: *"Until $B-F$ and $C-E$ traffic is in the zone below curve "A", the set of routing tables corresponding to "Curve A traffic (%)" should be used. If traffic is in zone I, then use the set of routing tables corresponding to "Curve B traffic (%)". If traffic is in zone II, then use the set of routing tables corresponding to "Curve C traffic (%)"*. These rules apply to the initial H that corresponds to $G = 425$.

A similar study can be executed with control vector "A" and the network in which the capacities of lines EF and FE were decreased to 42.63 pkts/sec. As a result a curve A_{opt} was plotted. The previous rule can be modified by adding the following: *"If $B-F$ and $C-E$ traffic is in the zone below the curve A_{opt}, then routing tables corresponding to "Curve A traffic (%)" should be used and the capacity of lines EF and FE can be decreased to 42.63 pkts/sec."*

Table 1: Network Traffic Routing

Source	Destination	Traffic (pkts/sec)	Path	Curve A traffic (%)	Curve B traffic (%)	Curve C traffic (%)
A	B	9.0	AB	100.0	100.0	100.0
A	C	4.0	ABC AEC	40.0 60.0	47.5 52.5	16.25 83.75
A	D	1.0	ABCD AECD ABFD	90.0 10.0	100.0	10.0 20.0 70.0
A	E	7.0	AE	100.0	100.0	100.0
A	F	4.0	AEF ABF	100.0	100.0	65.0 35.0
B	A	9.0	BA	100.0	100.0	100.0
B	C	8.0	BC	100.0	100.0	100.0
B	D	3.0	BCD BFD	50.3 49.7	100.0	89.7 10.3
B	E	2.0	BFE	100.0	100.0	100.0
B	F	*4.0 and vary*	BF	100.0	100.0	100.0
C	A	4.0	CBA CEA	51.25 48.75	32.5 67.5	71.25 28.75
C	B	8.0	CB	100.0	100.0	100.0
C	D	3.0	CD	100.0	100.0	100.0
C	E	*3.0 and vary*	CE	100.0	100.0	100.0
C	F	2.0	CEF	100.0	100.0	100.0
D	A	1.0	DCBA DCEA DFBA DFEA	30.0 60.0 10.0	60.0 5.0 35.0	5.0 60.0 35.0
D	B	3.0	DCB DFB	56.7 43.3	46.7 53.3	90.0 10.0
D	C	3.0	DC	100.0	100.0	100.0
D	E	3.0	DFE DCE	60.0 40.0	40.0 60.0	93.3 6.7
D	F	4.0	DF	100.0	100.0	100.0
E	A	7.0	EA	100.0	100.0	100.0
E	B	2.0	EFB	100.0	100.0	100.0
E	C	3.0	EC	100.0	100.0	100.0
E	D	3.0	EFD ECD	57.7 42.3	72.3 27.7	75.0 25.0
E	F	5.0	EF	100.0	100.0	100.0

Source	Destination	Traffic (pkts/sec)	Path	Curve A traffic (%)	Curve B traffic (%)	Curve C traffic (%)
F	A	4.0	FEA FBA	100.0	90.0 10.0	100.0
F	B	4.0	FB	100.0	100.0	100.0
F	C	2.0	FEC	100.0	100.0	100.0
F	D	4.0	FD	100.0	100.0	100.0
F	E	5.0	FE	100.0	100.0	100.0

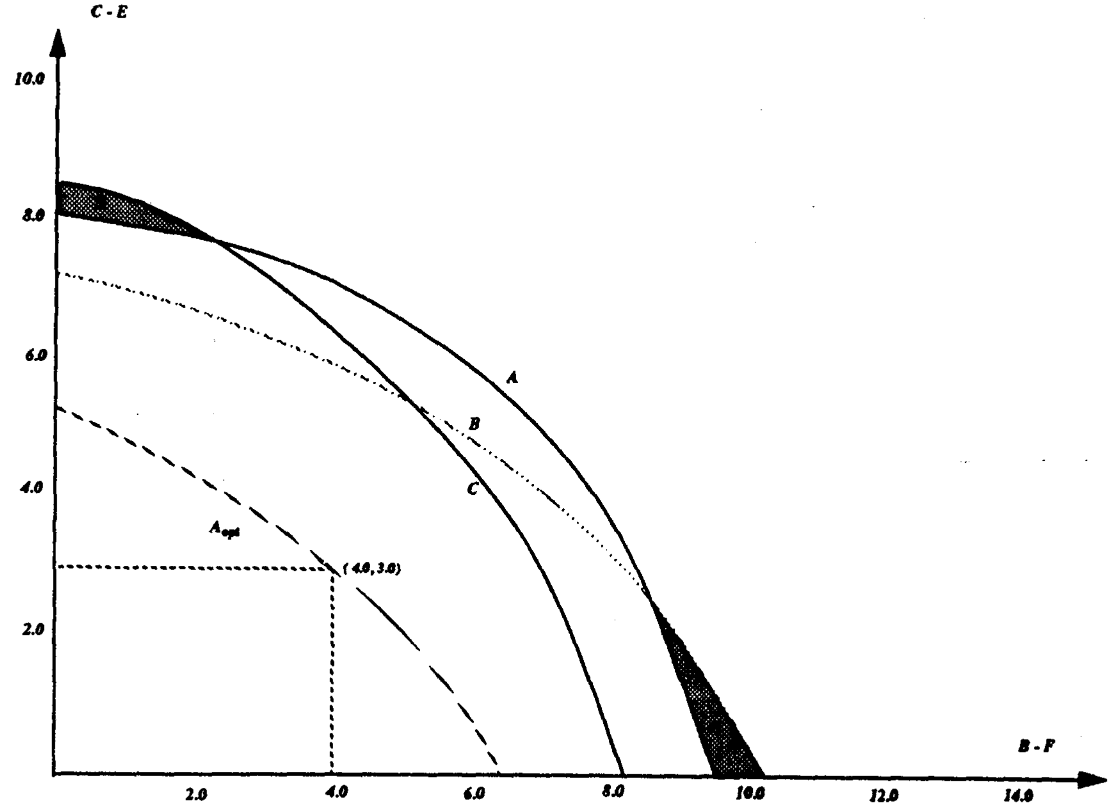


Figure 6

N93

29560

UNCLAS

A Neural Fuzzy Controller Learning by Fuzzy Error Propagation

Detlef Nauck Rudolf Kruse
Department of Computer Science
Technical University of Braunschweig
W-3300 Braunschweig, Germany

Keywords: fuzzy control, fuzzy error, fuzzy error propagation, membership function, learning algorithm, neural network

Abstract

In this paper we describe a procedure to integrate techniques for the adaptation of membership functions in a linguistic variable based fuzzy control environment by using neural network learning principles. This is an extension to our work in [2].

We solve this problem by defininig a fuzzy error that is propagated back through the architecture of our fuzzy controller. According to this fuzzy error and the strength of its antecedent each fuzzy rule determines its amount of error. Depending on the current state of the controlled system and the control action derived from the conclusion, each rule tunes the membership functions of its antecedent and its conclusion. By this we get an unsupervised learning technique that enables a fuzzy controller to adapt to a control task by knowing just about the global state and the fuzzy error.

1 Introduction

One of the design problems of a fuzzy controller is the choice of appropriate membership functions or the tuning of a priori membership functions in order to improve the performance of the fuzzy controller.

We solve this problem by defininig a fuzzy error that is propagated back through the neural-like architecture of our fuzzy controller. According to this fuzzy error, the strength of its antecedent, the current state of the controlled system, and the control action derived from the conclusion, each fuzzy rule determines its amount of error and tunes the membership functions of its antecedent and its conclusion. This paper is an extension to our work in [2], where we proposed a supervised learning algorithm depending on a non-fuzzy error.

We refrained from just integrating neural nets in certain parts of the architecture as black boxes as it is done in other approaches, or from adding an extra module to the architecture taking care of the correction of errors for example by weighting the rules according to the errors as it is described in [4, 12].

Our main concern is to keep the structure of the fuzzy controller that is determined by the fuzzy rules. We think of those rules as a piece of structural knowledge that gives us a roughly correct representation of the system to be controlled. If the actual output of the controller differs from the desired behaviour, we consider an unsuitable choice of membership functions that model the linguistic values of the system variables to be responsible [8].

We understand the adaptations of the membership functions as a reverse mechanism deduced from the forwarding inference machinery. We consider the computation of the control value from given measured input values as a feedforward procedure like in layered neural nets

[10], where the inputs are forwarded through the net resulting in some output values. If the actual output is not able to drive the controlled system to a desired state, an error has to be propagated back through the architecture changing parameters taking into account the feed forward propagation of inputs.

Because it is usually not possible to determine an optimal control action for a given state, we are not able to calculate the error of the produced output directly. This means we cannot use a supervised learning procedure like standard backpropagation. But by evaluating the state of the controlled system, we are able to determine a global error measure. This enables us to define a non-supervised learning algorithm. Training a fuzzy controller with such a learning procedure allows us to keep track of the changes and to interpret the modified rules.

The term "non-supervised" indicates in this context, that there is no "teacher" providing a desired output value to be compared to the actual output value. The controller is able to calculate the fuzzy error by just knowing about the state of the plant. From another point of view one could say, that the system is watched by a supervisor who uses "good" and "bad" signals to guide the learning procedure. But this kind of reinforcement learning is not considered to be a plain supervised procedure, and so we prefer to call our learning algorithm non-supervised, although it is derived from the BP-algorithm for neural networks [10].

Considering the ideas on which fuzzy controllers are based, we think it is a natural approach to use a fuzzy error for our system, which, according to its structure, we may call neural fuzzy controller.

In the following sections we first present the structure of our controller. Then we describe the fuzzy error propagation algorithm that we use as our learning procedure. Next we consider some simulation results concerning the control of an inverted pendulum and in the last section we discuss our results.

2 The Neural Fuzzy Controller

We consider a dynamical system S that can be controlled by one variable C and whose state can be described by n variables $X_1, \ldots, X_n$, i.e. we have a multiple input - single output system. For each of the mentioned variables we consider measurements in a subinterval $H = [h_1, h_2]$ of the real line. The imprecision is modelled by mappings $\mu : [h_1, h_2] \rightarrow [0, 1]$ in the sense of membership functions with the obvious interpretation as representations of linguistic values.

The control action that drives the system S to a desired state is described by the well-known concept of fuzzy if-then rules [13], where a conjunction of input variables associated with their respective linguistic values determine a linguistic value associated with the output variable. All rules are evaluated in parallel, and their outputs are combined to a fuzzy set which has to be defuzzified to receive the crisp output value. The conjunction of the inputs is usually done by the min-operation, and for the aggregation of the outputs of the rules the max-operation is usually chosen, as it is done by the well-known Zadeh-Mamdani procedure [7, 13].

For the evaluation of fuzzy rules the defuzzyfication-operation constitutes a problem that cannot be neglected. It is not obvious which crisp value is best suited to characterize the output fuzzy set of the rule system. In most of the fuzzy control environments the center-of-gravity method is used [5, 6]. Using this method, it is difficult to determine the individual part that each rule contributes to the final output value.

To overcome this problem we use Tsukamoto's monotonic membership functions, where the defuzzification is reduced to an application of the inverse function [1, 6]. Such a membership

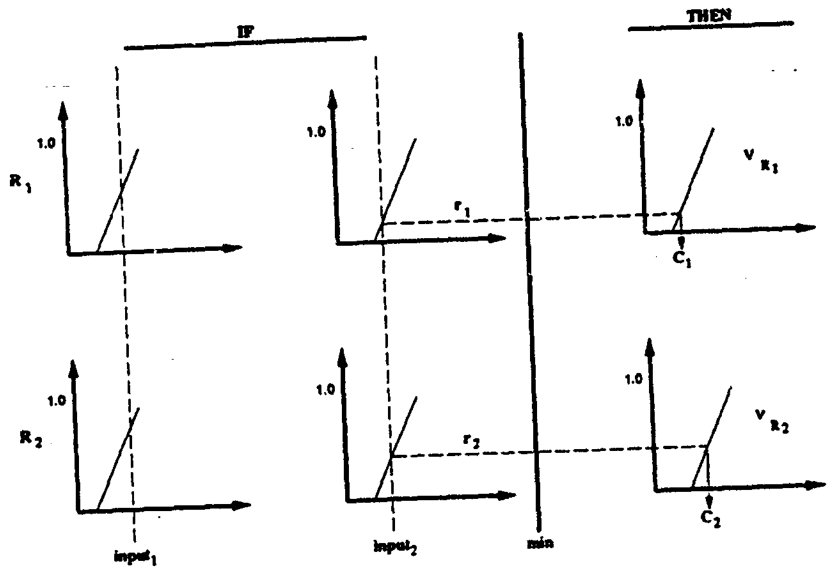


Figure 1: Defuzzification using Tsukamoto's monotonic membership functions

function μ is characterized by two points a, b with $\mu(a) = 0$ and $\mu(b) = 1$, and it is defined as

$$\mu(x) = \begin{cases} \frac{-x+a}{a-b} & \text{if } (x \in [a,b] \wedge a \leq b) \vee (x \in [b,a] \wedge a > b) \\ 0 & \text{otherwise} \end{cases}$$

The defuzzification is carried out by

$$x = \mu^{-1}(y) = -y(a-b) + a$$

with $y \in [0,1]$.

Consider the following two rules

R_1: **IF θ is PM AND $\dot{\theta}$ is PS THEN F is PS,**
R_2: **IF θ is PS AND $\dot{\theta}$ is PZ THEN F is PZ,**

where PM, PS and PZ represent the usual linguistic expressions *positive medium*, *positive small* and *positive zero*. The evaluation of those rules is presented in figure 1.

For our purposes we only need to restrict ourselves to monotonic membership functions to represent the linguistic values of the output variable. For the input variables the usual triangular, trapezoidal etc. membership functions can be chosen, even if we do not make use of this possibility in our controller for reasons of simplicity.

An example for the structure of our neural fuzzy controller is depicted in figure 2. The modules X_1 and X_2 represent the input variables that describe the state of the system to be controlled (plant, for short). These modules deliver their crisp values to their μ-modules which contain the membership functions interpreted as linguistic values assigned to the respective input variables. The μ-modules are connected to the following R-modules which represent the fuzzy if-then rules, the knowledge base of the controller. Each μ-module gives to its connected R-modules, the membership value $\mu_{ij}(x_i)$ of its input variable X_i. It is possible for each μ-module to be connected to several R-modules. The R-modules use a t-norm (min-operation

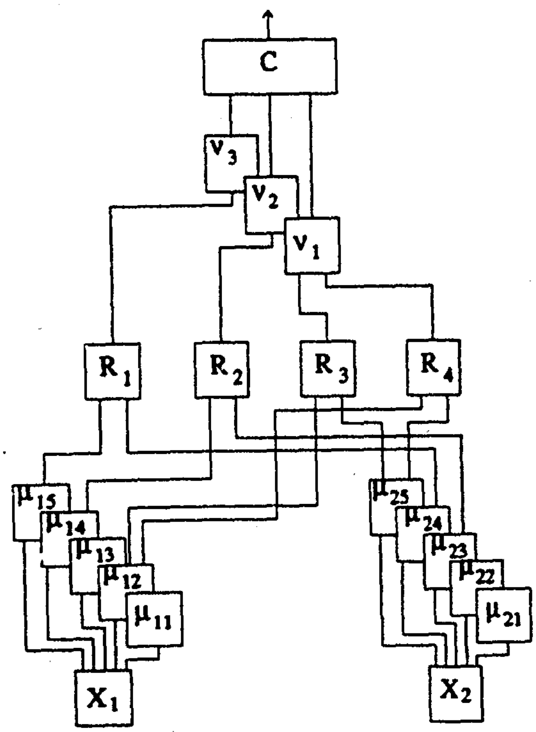


Figure 2: The structure of the neural fuzzy controller

in this case) to calculate the conjunction of their inputs and pass this value forward to one of the ν-modules, which contain the membership functions representing the linguistic values of the output variable. By passing through the ν-modules these values are changed to the conclusion of the respective rule. This means the implication (min-implication in this case) is carried out to obtain the value of the conclusion, which is usually a fuzzy set in a more general case. The conclusions are then passed on to the C-module where they are aggregated (e.g. by max-operation), and a crisp control value is determined by a defuzzification procedure.

In our case, however, monotonic membership functions are used, and so the ν-modules pass pairs $(r_i, \nu_k^{-1}(r_i))$ to the C-module, where the final output value is calculated by

$$c = \frac{\sum_{i=1}^{n} r_i \nu_{R_i}^{-1}(r_i)}{\sum_{i=1}^{n} r_i},$$

where n is the number of rules, and r_i is the degree to which rule R_i has fired.

From a more general point of view one can interpret the messages from the ν-modules to the C-module as fuzzy sets clipped by the min-implication at height r_i. The C-module aggregates the conclusions by a max-operation, and uses a non-standard defuzzification procedure as it is mentioned above.

As one can easily see, the system in figure 2 resembles a feedforward neural network. The X-, R-, and C-modules can be viewed as the neurons and the μ- and ν-units as the adaptable weights of the network. The X- and C-layer are identified as input layer, and output layer, respectively, and the R-layer serves as the intermediate or hidden layer that constitutes the internal representation of the network. The fact that one μ-module can be connected to more than one R-module is equivalent to connections in a neural network that share a common weight [9]. This is very important, because we want each linguistic value to be represented by only one membership function that is valid for all rules.

By this restriction we retain the structural knowledge that we put into the system by defining the rules. In other neural fuzzy systems this fact is not recognized [1, 3] and it is possible that one linguistic value is represented by different membership functions.

3 The Fuzzy Error Propagation Algorithm

Our goal is to tune the membership functions of the controller by a learning algorithm. Because it is usually not possible to calculate the optimal control action for a given state of the plant, so we can derive the error directly by comparing the optimal to the actual value, we are trying to obtain a measure that adequately describes the state of the plant under consideration.

The optimal state of the plant can be described by a vector of state variable values. That means, the plant has reached the desired state if all of its state variables have reached their value defined by this vector. But usually we are content with the current state if the variables have roughly taken these values. And so it is natural to define the goodness of the current state by a membership function from which we can derive a fuzzy error that characterizes the performance of our neural fuzzy controller.

Consider a system with n state variables $X_1, \ldots, X_n$. We define the fuzzy-goodness G_1 as

$$G_1 = \min(\mu_{X_1}^{optimal}, \ldots, \mu_{X_n}^{optimal}),$$

where the membership functions $\mu_{X_i}^{optimal}$ have to be defined according to the requirements of the plant under consideration.

In addition of a near optimal state we also consider states as good, where the incorrect values of the state variables compensate each other in a way, that the plant is driven towards its optimal state. We define the fuzzy-goodness G_2 as

$$G_2 = \min(\mu^{compensate_1}(X_1, \ldots, X_n), \ldots, \mu^{compensate_k}(X_1, \ldots, X_n))$$

where the membership functions $\mu^{compensate_j}$ again have to be defined according to the requirements of the plant. There may be more than one $\mu^{compensate_j}$ and they may depend on two or more of the state variables.

The overall fuzzy-goodness is defined as

$$G = g(G_1, G_2),$$

where the operation g has to be specified according to the actual application. In some cases a min-operation may be appropriate, and in other cases it may be more adequate to choose just one of the two goodness measures, perhaps depending on the sign of the current values of the state variables, e.g. we may want to use G_1 if all variables are positive or negative and G_2 if they are both positive and negative.

The fuzzy-error that is made by our neural fuzzy controller is defined as

$$E = 1 - G.$$

We are now able to define our learning algorithm that works for each fuzzy rule in parallel. Each rule R_i knows the value r_i of the conjunction of its antecedents and the value c_i of its conclusion. Because we are using monotonic membership funcions, c_i is already crisp. After the control action has been determined by the controller and the new state of the plant is known, we propagate the fuzzy-error E and the current values of the state variables to each R-module. If the rule has contributed to the control output, i.e. $r_i \neq 0$, it has to evaluate its own conclusion. According to the current state of the plant the rule can decide, whether its conclusion would drive the system to a better or to a worse state. The actual control value cannot be determined, but its direction, i.e. $\text{sgn}(c_{\text{opt}})$, is known. For the case $\text{sgn}(c_i) = \text{sgn}(c_{\text{opt}})$ the rule has to be made more sensitive and has to produce a conclusion that increases the current control action, i.e. makes it more positive or negative respectively. For the second case the opposite action has to be taken.

Consider that we are using Tsukamoto's monotonic membership functions. Each membership function can be characterized by a pair (a,b) such that $\mu(a) = 0$ and $\mu(b) = 1$ hold. A rule is made more sensitive by increasing the difference between these two values in each of its antecedents. That is done by keeping the value of b and changing a. That means the membership functions are keeping their positions determined by their b-values, and their ranges determined by $|a - b|$ are made wider. To make a rule less sensitive the ranges have to be made smaller. In addition to the changes in its antecedents, each firing rule has to change the membership function of its conclusion. If a rule has produced a good control value, this value is made better by decreasing the difference $|a - b|$, and a bad control value is made less worse by increasing $|a - b|$.

The rules change the membership functions by propagating their own rule-error

$$e_{R_i} = \begin{cases} -r_i \cdot E & \text{if } \text{sgn}(c_i) = \text{sgn}(c_{\text{opt}}) \\ r_i \cdot E & \text{if } \text{sgn}(c_i) \neq \text{sgn}(c_{\text{opt}}) \end{cases}$$

to the connected μ- and ν-modules. The changes in the membership functions of the conclusions (ν-modules) are calculated according to

$$a_k^{\text{new}} = \begin{cases} a_k - \sigma \cdot e_{R_i} \cdot |a_k - b_k| & \text{if } (a_k < b_k) \\ a_k + \sigma \cdot e_{R_i} \cdot |a_k - b_k| & \text{otherwise,} \end{cases}$$

where σ is a learning factor and R-module R_i is connected through ν_k to the C-module. If ν_k is shared, it is changed by as much R-modules as are connected to the C-module through this membership function. For the membership functions of the antecedents (μ-modules) the follwing calculation is carried out:

$$a_{jk_j}^{\text{new}} = \begin{cases} a_{jk_j} + \sigma \cdot e_{R_i} \cdot |a_{jk_j} - b_{jk_j}| & \text{if } (a_{jk_j} < b_{jk_j}) \\ a_{jk_j} - \sigma \cdot e_{R_i} \cdot |a_{jk_j} - b_{jk_j}| & \text{otherwise,} \end{cases}$$

where the X-module X_j is connected to the R-module R_i through the membership function μ_{jk_j}, with $k_j \in \{1, \ldots, s_j\}$, and s_j is the number of linguistic values of X_j. If μ_{jk_j} is shared, it is changed by as much R-modules as X_j is connected to through this μ-module.

Compared to learning algorithms used in neural networks one can see, that the error is not just passed back through the system, but that it is propagated to the intermediate layer constituted by the R-modules, where a rather sophisticated evaluation of this error is carried out, which is not typical for connectionistic systems. There the error signal is treated equally by each component of the network. In our system the R-modules propagate the error back and forward to the μ- and ν-modules, respectively, where less complicated calculations lead to a change of the membership functions, the "fuzzy weights" from a connectionistic point of view.

The neural fuzzy controller has not to learn from scratch, but knowledge in the form of fuzzy if-then rules can be coded into the system. The learning procedure does not change this structural knowledge. It tunes the membership functions in an obvious way, and the semantics of the rules are not blurred by any semantically suspicious factors or weights attached to the rules.

4 Controlling an Inverted Pendulum

In this section we present the results of a simulation of the neural fuzzy controller applied to the inverted pendulum (figure 4). An inverted pendulum is a well-known nonlinear dynamical system, often used to test fuzzy controllers. The sytem is described by two state variables that are the input variables to the controller, the angle θ measured against the upright position and the angle velocity $\dot{\theta}$, which also describe the error and the change of error. The pendulum is controlled by one control variable that is the control output, the force F applied to the base of the rod that is only allowed to fall to the left or to the right. We use a simplified version of the inverted pendulum in our simulation. The system is described by the differential equation

$$(m + \sin^2\theta)\ddot{\theta} + \frac{1}{2}\dot{\theta}^2 \sin(2\theta) - (m+1)\sin\theta = -F\cos\theta.$$

The movement of the rod is simulated by a Runge-Kutta procedure with a timestepwidth of 0.1.

There are eight linguistic values attributed to each of the three variables. This are the common values PL, PM, PS, PZ, NZ, NS, NM, NL. Because we use monotonic membership-functions that are not symmetric, we modell the value Zero as PositiveZero and NegativeZero. A typical definition for the membership functions used in our simulation is depicted in figure 3. The exact initial values of the above mentioned characterizing point b with $\mu(b) = 1$ and the range d defined as $b - a$ with $\mu(a) = 0$ for each membership function can be found in the following tables.

The rule base used to construct the controller is presented in figure 5. It was found that the controller was already able to balance the pendulum with this rule base and the initial membership functions. However, after the learning procedure was activated, the performance of the system became much better. It was also found that the controller was not able to cope with extreme initial positions of the pendulum, e.g. $\theta = 20$ and $\dot{\theta} = 2$, when only the initial

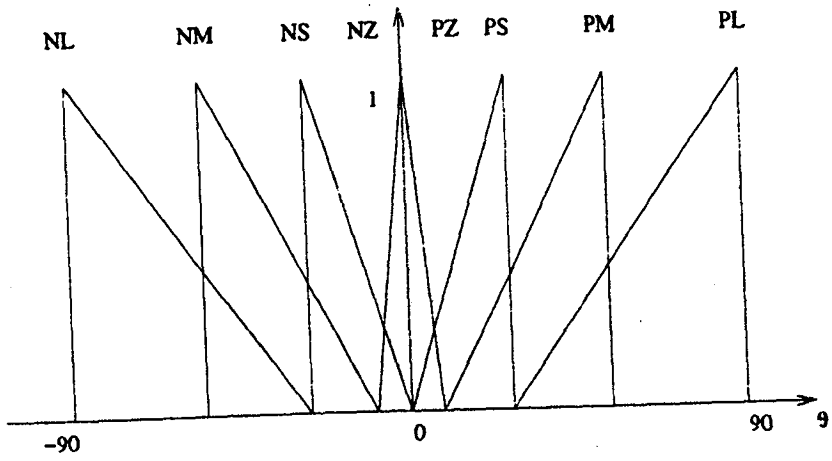


Figure 3: Monotonic membership functions modelling the linguistic values of θ

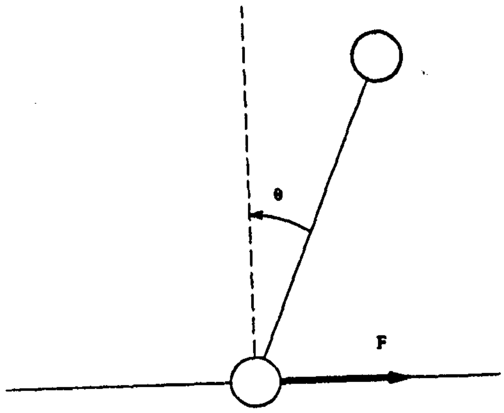


Figure 4: The inverted pendulum

$\dot{\theta}$

		PL	PM	PS	PZ	NZ	NS	NM	NL
	PL	PL							
	PM		PM	PS	PS				
	PS		PS	PS	PS				
θ	PZ			PS	PZ				
	NZ					NZ	NS		
	NS					NS	NS	NS	
	NM					NS	NS	NM	
	NL								NL

Figure 5: The rule base of the neural fuzzy controller

membership functions were used. But when the learning algorithm was applied the controller was able to balance the rod after only three trials, i.e. the pendulum fell down ($|\theta| > 90$) just twice.

The results of the simulation can be found in table 1. We have only documented the changes in the membership functions of θ. For the other two variables similar changes have been found.

The fuzzy error E has been defined by

$$E = \begin{cases} 1 - \min(1 - \frac{|\theta|}{3}, 1 - \frac{|\dot{\theta}|}{0.3}) & \text{if } \operatorname{sgn}(\theta) = \operatorname{sgn}(\dot{\theta}),\ -3 \leq \theta \leq 3,\ -0.3 \leq \dot{\theta} \leq 0.3 \\ \frac{|\theta + 10\,\dot{\theta}|}{3} & \text{if } \operatorname{sgn}(\theta) \neq \operatorname{sgn}(\dot{\theta}),\ -3 \leq \theta + 10\,\dot{\theta} \leq 3 \\ 1 & \text{otherwise.} \end{cases}$$

That means the fuzzy error is defined by one two-dimensional and two one-dimensional membership functions. The learning rate σ has been set to 0.01.

The next table shows the performance of the controller for each of the five runs. Each run consisted of 3000 loops and has been performed with and without the learning procedure. The performance is measure by the average over all absoulte values of θ measured during a run with

			Run 1	Run 2	Run 3	Run 4	Run 5
	b	initial d	$\theta = 0$ $\dot{\theta} = 0.1$	$\theta = 10$ $\dot{\theta} = 1$	$\theta = 20$ $\dot{\theta} = 1$	$\theta = 20$ $\dot{\theta} = 2$	$\theta = 30$ $\dot{\theta} = 2$
NL	-90	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000
NM	-70	60.0000	60.0000	60.0000	60.0000	60.0000	171.9157
NS	-40	40.0000	48.9108	47.5219	49.0139	64.3483	74.5797
NZ	0	-13.0000	-8.0298	-9.5576	-10.0961	-9.6819	-9.2074
PZ	0	13.0000	31.4427	17.6432	13.5961	13.6028	15.1134
PS	40	-40.0000	-49.8010	-48.2032	-50.0540	-47.7372	-60.5255
PM	70	-60.0000	-60.0000	-61.0686	-62.0523	-82.9168	-180.7220
PL	90	-60.0000	-60.0000	-60.0000	-60.0000	-60.0000	-60.0000

Table 1: The changes in the ranges of the membership functions of θ

	run 1	run 2	run 3	run 4	run 5
$\bar{\theta}$ without learning	4.05	4.17	4.29	n.a.t.b.	n.a.t.b.
$\bar{\theta}$ with learning	0.56	0.65	0.74	2.13	5.36
no. of trials	1	1	1	3	11

Table 2: The performance of the controller

n loops:

$$\bar{\theta} = \sum_{i=1}^{n} \frac{|\theta|}{n}.$$

The controller was able to keep an angle near zero with activated learning procedure, and was also able to balance the pendulum beginning from the extreme positions of runs 4 and 5 in 3 and 11 trials, respectively, whereas the controller was not able to balance (n.a.t.b.) the rod in these cases without learning.

5 Discussion

We have presented a learning algorithm for a neural fuzzy controller based on a fuzzy error measure. The structure of the controller resembles a neural network and the fuzzy error propagation can be compared to non-supervised learning procedures as they exist for certain kinds of connectionistic systems. Simulations of the controller have shown that the learning procedure improves the behaviour of the fuzzy controller and is able to handle situations where the non-learning controller fails.

The introduced fuzzy error measure is suitable for describing the performance of the controller and allows each rule to determine changes for the membership functions of its preconditions and its conclusion. The learning algorithm starts from a predefined rule base that can be obtained by clustering methods e.g. [11], and it does not change the structural knowledge encoded in these rules. It leaves the semantics of each rule intended by the user unchanged, but removes the errors caused by an inaccurate modelling by changing the fuzzy sets. The results of the learning procedure can be easily interpreted [8]. It is not possible that two rules use different fuzzy sets describing the same linguistic value.

Other neural fuzzy control environments, which are based more on neural network architectures [1, 3], often use factors to weight the rules or allow the rules to have different representations for the same input value. From our point of view in this case there are semantics involved, that are different to our approach. This has to be considered when an adaptive fuzzy control environmemt is used.

An extension to the presented learning algorithm that is used in combination with a fuzzy neural network that is capable to learn fuzzy rules and membership functions is presented in [9].

References

[1] Hamid R. Berenji: A Reinforcement Learning-Based Architecture for Fuzzy Logic Control. Int. J. Approximate Reasoning, vol. 6, no. 2, 267-292 (1992)

[2] P. Eklund, F. Klawonn, D. Nauck: Distributing Errors in Neural Fuzzy Control. Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA-92), 1139-1142, (1992)

[3] Jyh-Shing R. Jang: Fuzzy Modelling Using Generalized Neural Networks and Kalman Filter Algorithm. Proc. AAAI-91, vol. 2, 762-767, (1991)

[4] B. Kosko: Neural Networks and Fuzzy Systems. Prentice–Hall, Englewood Cliffs (1992)

[5] C.C. Lee: Fuzzy Logic Control Systems: Fuzzy Logic Controller - Part I. IEEE Trans. Syst. Man Cybern., vol. 20, no. 2, 404-418, (1990)

[6] C.C. Lee: Fuzzy Logic Control Systems: Fuzzy Logic Controller - Part II. IEEE Trans. Syst. Man Cybern., vol. 20, no. 2, 419-435, (1990)

[7] E.H. Mamdani: Applications of Fuzzy Algorithms for Simple Dynamic Plant. Proc. IEE vol. 121, 1585-1588 (1974)

[8] D. Nauck, R. Kruse: Interpreting Changes in the Membership Functions of a Self–Adaptive Neural Fuzzy Controller. Proc. 2nd Int. Workshop on Industrial Applications of Fuzzy Control and Intelligent Systems, Dec. 2-4, 1992, College Station, Texas, (1992)

[9] D. Nauck, R. Kruse: A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation. Submitted for publication, (1992)

[10] D.E. Rumelhart, J.L. McClelland: Parallel Distributed Processing, Vol. 1. MIT Press, Cambridge, Massachusetts (1986)

[11] H. Takagi, I. Hayashi: NN–driven Fuzzy Reasoning. Int. J. Approximate Reasoning vol. 5 no. 3, 191-212 (1991)

[12] S. Shao: Fuzzy Self-Organizing Controller and its Application for Dynamic Processes. Fuzzy Sets and Systems 26, 151-164 (1988)

[13] L.A. Zadeh: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans. Syst. Man Cybern., vol. SMC–3, 28–44 (1973)

DETERMINING RULES FOR CLOSING CUSTOMER SERVICE CENTERS: A PUBLIC UTILITY COMPANY'S FUZZY DECISION

André de Korvin
Department of Applied Mathematical Sciences
University of Houston-Downtown

Margaret F. Shipley
Department of Business Management
University of Houston-Downtown
One Main Street
Houston, Texas 77002

Robert N. Lea
Lyndon B. Johnson Space Center
Houston, Texas 77058

Abstract

In the present work, we consider the general problem of knowledge acquisition under uncertainty. Simply stated, the problem becomes: how can we capture the knowledge of an expert when the expert is unable to clearly formulate how he or she arrives at a decision?

A commonly used method is to learn by examples. We observe how the expert solves specific cases and from this infer some rules by which the decision may have been made. Unique to our work is the fuzzy set representation of the conditions or attributes upon which the expert may possibly base his fuzzy decision. From our examples, we infer certain and possible fuzzy rules for closing a customer service center and illustrate the importance of having the decision closely relate to the conditions under consideration.

1. Introduction

Much effort has recently been devoted to studying the problem of knowledge acquisition under uncertainty. Uncertainty arises in many different situations. It may be caused by the ambiguity in the terms used to describe a specific situation. It may also be caused by skepticism of rules used to describe a course of action or by missing and/or erroneous data. [See (Arciszewski & Ziarko 1986), (Bobrow, et.al. 1986), (Wiederhold, et. al. 1986), and (Zadeh 1983).]

To deal with uncertainty, techniques other than classical logic and the application of statistical methods need to be developed. [See Mamdani, et. al. (1985) for a study of the limitations of traditional statistical methods.] Rough set theory can address the limitations of statistics in dealing with uncertainty while allowing rules to be extracted that describe a course of action or a decision to be made. [See (Fibak, et. al. 1986), (Grzymala-Busse 1988), (Mrozek 1985 & 1987), (Pawlak 1981, 1982, 1983 & 1985), and (Arciszewski & Ziarko 1986).] Fuzzy set theory is another tool used to deal with uncertainty where ambiguous terms are present. [See (Zadeh 1979, 1981 & 1983)] Our work builds on these alternatives to statistics, allowing us to infer knowledge from the uncertainty associated with ambiguous (i.e. fuzzy) terms.

2. Development of the Model

The main purpose of the present work is to study the general situation where the decision maker is faced with uncertain (i.e. fuzzy) conditions and makes a fuzzy decision which might be strongly or weakly based on these conditions. In this situation, fuzzy rules can be extracted. We shall present the basic notations and concepts for developing a methodology to extract such rules from fuzzy conditions and fuzzy decisions. [Most of these concepts are discussed in (Grzymala-Busse 1988), and (Pawlak 1981, 1982 & 1985) as they relate to crisp sets.]

Basic Notations and Concepts

Let U be the universe. Let R be an equivalence relation on U. Let X be any subset of U. If [x] denotes the equivalence class of x relative to R, then we define

$\underline{R}(X) = \{x \in U/[x] \subset X\}$ and

$\overline{R}(X) = \{x \in U/[x] \cap X \neq \emptyset\}$.

$\underline{R}(X)$ is called the lower approximation of X and $\overline{R}(X)$ is called an upper approximation of X. Then $\underline{R}(X) \subset X \subset \overline{R}(X)$. If $\underline{R}(X) = X = \overline{R}(X)$, then X is called definable.

An information system is a quadruple (U,Q,V,τ) where U is the universe and Q equals $C \cup D$ where $C \cap D = \emptyset$. The set C is called the set of conditions (attributes); D is called the set of decisions. The set V stands for value and τ is a function from UxQ into V where $\tau(u,q)$ denotes the value of condition q for element u. The set C induces naturally an equivalence on U by partitioning U into sets over which all attributes are constant. The set X is called roughly C-definable

if $\underline{R}(X) \neq \emptyset$ and $\overline{R}(X) \neq U$. It will be called externally C-definable

if $\underline{R}(X) = \emptyset$ and $\overline{R}(X) \neq U$. It will be called internally C-definable

if $\underline{R}(X) \neq \emptyset$ and $\overline{R}(X) = U$.

Unfortunately, uncertainty is all too often present in the conditions and the decisions. The conditions and the decisions fail to partition the universe into well-defined classes and some overlap is present. We will deal with the issue of using rough set theory to handle the lack of clearly differentiated partitions by using fuzzy sets. We will thus need to "fuzzify" rough set theory.

Rough Set Notation Applied to Fuzzy Sets

Two functions on pairs of fuzzy sets that will be used to determine rules for closing a utility company's customer service centers (CSCs). We define:

$$I(A \subset B) = \inf_{x} \text{Max } \{1 - A(x), B(x)\} \qquad (1)$$

$$J(A \# B) = \text{Max}_{x} \text{ Min } \{A(x), B(x)\}. \qquad (2)$$

Here A and B denote fuzzy subsets of the same universe. The function $I(A \subset B)$ measures the degree to which A is included in B and $J(A \# B)$ measures the degree to which A intersects B. It is clear that I and J can be expressed as

$$I(A \subset B) = \inf_x (A \rightarrow B) \tag{3}$$

$$J(A \# B) = \operatorname{Max}_x (A \cap B). \tag{4}$$

In addition, the following relation holds:

$$I(A \subset B) = 1 - J(A \# \neg B). \tag{5}$$

We can define the fuzzy terms involved in the decision as a function of the terms used in the conditions. Let $\{B_i\}$ be a finite family of fuzzy sets. Let A be a fuzzy set. By a lower approximation of A through $\{B_i\}$, we mean the fuzzy set

$$\underline{R}(A) = \bigcup_i I(B_i \subset A) B_i \tag{6}$$

The decision making process may be simplified by disregarding all sets B, if $I(B_i \subset A)$ is less than some threshold α. Then,

$$\underline{R}(A)_\alpha = \bigcup_i I(B_i \subset A) B_i \tag{7}$$

over all B_i for which $I(B_i \subset A) \geq \alpha$. Similarly, we can define the upper approximation of A through $\{B_i\}$ as

$$\overline{R}(A)_\alpha = \bigcup_i J(B_i \# A) B_i \tag{8}$$

over all B_i for which $J(B_i \# A) \geq \alpha$.

The operators I and J will yield two possible sets of rules: the certain rules and the possible rules. The data given for the Customer Service Centers (CSCs) will be converted to fuzzy diagnosis of the attributes and we will be able to extract fuzzy rules from the raw data. Each rule for the decision to close a CSC will have some measure of belief associated with it. The primary objective is to see to what degree a combination of attributes is a subset of the decision (certain rules) or intersects the decision (possible rules) to close a customer service center. In addition, fuzzy terms involved in the decision have a lower and an upper approximation so that we have a measure of the minimum degree to which the lower approximation implies the decision and the minimum degree to which that decision satisfies the upper approximation.

It is important to realize that the present methodology does not give any indication of the quality of the decision. What is determined is how closely the decision maker seems to depend on the values of the selected set of attributes. If the decisions seem to follow consistently these values and if we trust the decision maker, we then have acquired knowledge, in terms of these attributes, as to how decisions are made.

3. Application

Houston Lighting & Power Company is the largest investor-owned electric utility in the Southwest. HL&P is responsible for generating and distributing electricity throughout twelve counties surrounding Houston. Even though it is a private company, its operations are regulated in Texas by the Public Utility Commission (PUC).

In November 1988, HL&P filed a request with the Public Utility Commission for a \$432 million rate increase. The public's perception of HL&P's stability and sound judgment in the daily management of its operations was critical to the outcome of the rate case. HL&P needed to show that its decisions and operating procedures were initiated with total consideration given to effectively serving its customers.

However, the company's management felt that in order to reduce operating expenses in the event that the rate request before the PUC was denied one or more Customer Service Centers (CSCs) might have to be closed. These customer service centers handled walk-in customer traffic for payment of bills and general customer inquiries and, thus, were operated for the public's convenience. With the rate increase request before the PUC, HL&P had to carefully analyze the CSC closing decision. The main consideration for HL&P was the public's reaction. Although a decision to close a site would potentially impact only a few customers, there might be those who challenged the PUC rate hike request on the grounds of paying more for less service.

HL&P investigated all relevant factors in making its decision. The difference in relative operating expenses of CSCs was negligible according to the company's operating and maintenance budget. Therefore, operating cost could not be regarded as a major consideration in the elimination of one of the CSCs. Four factors could be considered in this decision: the total number of customers in a district, the increase or decrease in a district's population, the number of customers utilizing the CSC in relation to the district's population, and the distance that customers would have to travel to an alternate CSC in the event their local CSC was closed. (See Table 1.)

TABLE 1: Customer Service Center Data

	Avg. Customers in District	% Change in Customers	Usage/ Population	Rerouting Miles
Bayshore	38,510	5.1	4.64	15
Baytown	36,360	-1.4	21.5	15
Brazoria	20,689	3.4	14.07	20
Brazosport	21,976	.4	8.51	20
Cypress	44,074	8.3	1.87	17
Fort Bend	39,145	5.3	15.5	18
Galveston	31,263	- .1	36.44	20
Humble	55,911	1.0	12.44	15
Katy/Sealy	26,760	2.4	18.54	17
Wharton	8,707	- .74	39.43	18

NOTE: All of the above is based on 1985-1987 data.

Based upon the data given in Table 1, one of the authors served as a decision maker in specifying a value indicative of a high number of customers in the district and a low number in the district; a great and a small percent change in usage; a high and a low percentage of customers utilizing the center; and a large and small rerouting distance. A high number of customers was 60,000 and a low number of customers was 5000 A great percent change was ± 9.00 and a small percent change was ± 0.1. A high usage population ratio was 40.00 percent and a low usage was 1.00 percent. A large rerouting distance was 20 miles and a small distance was 10 miles. The degree to which each site satisfied the definition of high, low; great, small; high, low; and large, small is given by dividing the actual data given in Table 1 by the parameter values defined above to yield those values given in Table 2.

TABLE 2: Values for Fuzzy Sets of Conditions

	Avg.Customers in District		% Change in Customers		Usage/ Population		Rerouting Miles	
	HIGH	LOW	GREAT	SMALL	HIGH	LOW	LARGE	SMALL
Bayshore	.640	.130	.567	.020	.116	.216	.75	.667
Baytown	.606	.138	.156	.071	.538	.047	.75	.667
Brazoria	.345	.242	.378	.029	.352	.071	1.00	.500
Brazosport	.366	.228	.044	.250	.213	.118	1.00	.500
Cypress	.735	.113	.922	.012	.047	.535	.85	.588
Fort Bend	.652	.128	.589	.019	.388	.065	.90	.556
Galveston	.521	.160	.011	1.000	.911	.027	1.00	.500
Humble	.932	.089	.111	.100	.311	.080	.75	.667
Katy/Sealy	.446	.187	.267	.042	.464	.054	.85	.588
Wharton	.145	.574	.082	.135	.986	.025	.90	.556

Using the total operating revenue generated for each service center, our decision maker determined that if revenue was less that 1% of the total generated from all centers, the CSC would be closed. Conversely, the center would not be closed if revenue exceeded 10% of the total. The raw data and the reflective valuation of each center for closing and not closing are given in Table 3.

TABLE 3: Revenue of each CSC & Closing Weight

	Total Dollar Revenue	Close	Do Not Close
Bayshore	270,411,636	.039	1.000
Baytown	142,262,298	.075	1.000
Brazoria	44,464,243	.239	.419
Brazosport	144,290,786	.074	1.000
Cypress	92,178,304	.115	.869
Fort Bend	88,498,221	.120	.834
Galveston	89,125,871	.119	.840
Humble	120,219,083	.088	1.000
Katy/Sealy	53,675,510	.198	.506
Wharton	15,660,308	.677	.148
	1,060,786,260		

Of course, no one at HL&P would specifically state exactly how the decision to close a CSC would be determined. Since most businesses define profitability in terms of revenue generated and since HL&P representatives had obtained this information, we have assumed that thr total operating revenue would be the major factor affecting the decision to close a CSC.

In reality, many factors, some of them even unknown to the decision maker himself, may impact the decision of closing a Customer Service Center. Still, we are interested in learning by examples how much the decision can be attributed to the attributes for which HL&P had accumulated data for each CSC.

Example 1

In the first example we selected two attributes: Usage/Population and Rerouting Distance.

First, we let x_i denote the customer service centers, such that x_1 = Bayshore, x_2 = Baytown,..., x_{10} = Wharton. Then D_A = Close the CSC, and D_B = Do Not Close the CSC. The decision to close the facility can be evaluated as:

$D_A = .039/X_1 + .075/x_2 + .239/x_3 + .074/X_4 + .115/x_5 + .120/x_6$
$+ .119/x_7 + .088/x_8 + .198/x_9 + .677/x_{10}$

This indicates that based upon revenue generated, Wharton is a fairly good example of a CSC to be closed, while Bayshore is not a good example of D_A.

Likewise, we can indicate the degree of membership of each CSC for each fuzzy-defined condition/attribute: High (H) Usage/Population, Low (L) Usage/Population, Large (G) Rerouting Distance, and Small (S) Rerouting Distance. For example, we define the fuzzy set H as:

$H = .116/x_1 + .538/x_2 + .352/x_3 + .213/x_4 + .047/x_5 + .388/x_6 +$
$.911/x_7 + .311/x_8 + .464/x_9 + .986/x_{10}$

We compute the minimum degree to which possible combinations of conditions/attributes are related to decision D_A. Thus,

$I (H \subset D_A) = .119$	$I (H \cap G \subset D_A) = .119$
$I (L \subset D_A) = .465$	$I (H \cap S \subset D_A) = .462$
$I (G \subset D_A) = .074$	$I (L \cap G \subset D_A) = .465$
$I (S \subset D_A) = .333$	$I (L \cap S \subset D_A) = .465$

With a threshold of $\alpha = 0.40$, the rules for closing a CSC are:

1. If usage/population percentage is low (i.e. 1% or less of the customers in the district utilizing the CSC), then the CSC should be closed. (D_A is present .465 or Belief = .465)
2. If the usage/population percent is high (approximately 40% of the customers in the district utilize the CSC) and the rerouting distance is small (approximately 10 miles), then the CSC should be closed. (Belief = .462)
3. If the usage/population percent is low and the rerouting distance is high (20 miles), then the CSC should be closed. (Belief = .465)
4. If the usage/population is low and the rerouting distance is low, the CSC should be closed. (Belief = .465)

Since no new information is provided by rules 3 and 4, the extracted rules for closing are:

1. If usage/population percentage is low then the CSC should be closed. [The belief is .465.]
2. If usage/population is high and the rerouting distance is small then the CSC should be closed. [The belief is .462.]

Rule 1 is certainly reasonable. Rule 2 sounds less reasonable. It is generated by the decision maker deciding fairly strongly in favor of Wharton to be closed, although its usage/population was definitely high and its rerouting distance was over .5 small. From such examples, we learn that for high usage and relatively low rerouting distance a CSC can be closed. Note that from the data, we do not feel that strongly about these rules. The extracted ıules would not be sufficient to infer closing from past experience.

We now measure the degree to which the fuzzy sets intersect D_A as:

$J (H \# D_A) = .677$ $J (H \cap G \# D_A) = .677$
$J (L \# D_A) = .115$ $J (H \cap S \# D_A) = .556$
$J (G \# D_A) = .677$ $J (L \cap G \# D_A) = .115$
$J (S \# D_A) = .556$ $J (L \cap S \# D_A) = .115$

With $\alpha = 0.60$, the acceptable rules are:

5. If usage/population percent is high, then closing is possible .667.
6. If rerouting distance is great, then closing is possible .677.
7. If usage/population is high and rerouting distance is great, then closing is possible .677.

The extracted rule would be Rule 7. The possibility of closing if usage/population is high and rerouting distance is great can't be discounted. Brazoria was recommended to be closed with strength .239 versus not closing with strength .419. Nevertheless, the rerouting distance was definitely high and the usage/population was rated .352 high versus .071 low.

We determine the lower approximation of D_A, using α =.40, as:

$$\underline{R} (D_A) = .465 L \cup .465 (L \cap G) \cup .465 (L \cap S) \cup .462 (H \cap S)$$
$$= .465 L \cup .462 (H \cap S)$$

Note that this result shows Rule 3 and Rule 4 to be superfluous to Rule 1 and unnecessary for the calculation of $\underline{R} (D_A)$.

We can also show that Rules 5 & 6 should not be accepted since the upper approximation of D_A for $\alpha = .60$, results in Rule 7:

$$\overline{R} (D_A) = .677 H \cup .677 G \cup .677 (H \cap G)$$
$$= .677 H \cup .677 G$$

Although, Rule 1 appears to be the most logical rule to accept, it eliminates Wharton as the primary candidate for closing. It should be noted that Wharton's valuative scores based on high customer utilization (.986) and relatively large as well as relatively small rerouting values (.90 and .556, respectively) are influencing the second and third decision rules. This example is an excellent illustration of the necessity for the attributes to properly reflect the decision criteria. In this example, the decision to close a center was to be based solely on revenue generated. This means that HL&P would select a center which generated the lowest revenue as that to be closed and the one which generated the highest revenue becomes that least likely to be closed. This suggests that Wharton is our best site to close. However, the usage/population percentage at Wharton is high leading one to the conclusion that, in general, those centers with high customer usage should be closed.

Example 2

A second example is given to show that a closer relationship between the decision and the attributes selected will lead to seemingly more logical rules being determined. For this illustration, we used the size of the customer base with the percent usage which suggests that although the percent usage may be high, there may be many fewer customers at the center generating much less revenue, thus making the center a candidate for closing.

Using the values of the fuzzy sets High (NH) and Low (NL) for the number of customers, and High (UH) and Low (UL) for the usage/population percentages given in Table 2:

$I (NH \subset D_A) = .088$ $\quad I (NH \cap UH \subset D_A) = .463$
$I (NL \subset D_A) = .677$ $\quad I (NH \cap UL \subset D_A) = .465$
$I (UH \subset D_A) = .119$ $\quad I (NL \cap UH \subset D_A) = .677$
$I (UL \subset D_A) = .465$ $\quad I (NL \cap UL \subset D_A) = .87$

With $\alpha = .60$, the following rules would be determined:

1. If the number of customers is low, the belief that the CSC should be closed is .677.
2. If the number of customers is low and the usage/population is low, the CSC should be closed .87.
3. If the number of customers is low and the usage/population is high, the CSC should be closed .677.

Rule 3 is redundant and we would keep Rules 1 and 2.

Also using $\alpha = .60$, we can determine the following rules from:

$J (NH \# D_A) = .239$ $\quad J (NH \cap UH \# D_A) = .239$
$J (NL \# D_A) = .574$ $\quad J (NH \cap UL \# D_A) = .115$
$J (UH \# D_A) = .677$ $\quad J (NL \cap UH \# D_A) = .574$
$J (UL \# D_A) = .116$ $\quad J (NL \cap UL \# D_A) = .113$

4. If the number of customers in the district is low, closing is possible .574.
5. If the usage/population is high, closing is possible .677.
6. If the number of customers in the district is low and the usage/population is high, closing is possible .574.

From these rules, we select Rule 5.

Computing the upper and lower approximations based on $\alpha = .60$, we have:

$\underline{R} (D_A) = .677 \text{ NL} \cup .87 (\text{NL} \cap \text{UL}) \cup .677 (\text{NL} \cap \text{UH})$ and

$\overline{R} (D_A) = .677 \text{ UH}$ such that:

Thus, the acceptable rules where Rule 1 and Rule 2 come from certainty and Rule 3 come from possibility are:

1. If the number of customers is low and usage/population is low, the CSC should be closed. [Belief is .87.]
2. If the number of customers is low, the CSC should be closed. [Belief is .677.]
3. If the usage/population is high, the CSC can be closed. [Plausibility is .677.]

If strictly ordering the CSCs to be closed based upon Rule 2, Wharton would be the decision maker's first choice for closing (followed by Brazoria and Brazosport). Although Rule 3 appears to be illogical, if strictly ordering a center to be closed based upon this rule, Wharton would be selected (followed by Galveston and Baytown). If using the more logical Rule 1, Wharton would not be considered first. Brazoria, ranking second in having the lowest number of customers and fifth in having a low usage/population ratio would be one possible choice for a CSC to be closed. Brazosport with the third lowest number of customers and the third lowest usage/population ratio would also be a good choice for closure. Notice that these were the second choices if strictly ordering by Rule 2, based upon the number of customers in the district. Since the number of customers in the district would directly relate to the revenue generating power of a CSC, this example provides a more realistic result and supports the need to have well chosen attributes, reflecting the decisions made.

4. Conclusions

Since the crisp set is a limiting case of the fuzzy setting, expected benefits that arise from our fuzzy set based method are a more realistic and general approach to knowledge acquisition. Acquisition of knowledge through examples, which is particularly of interest when the decision maker is unable to articulate how he arrives at a decision, is a very natural approach to learning. Our process allows the user to learn and determine rules based on the examples available. Of course, the quality of the learning depends upon the relevance of the chosen attributes to the decision.

The process allows rules to be determined through incorporation of attribute data for all available alternatives for which a decision must be made. The decision maker can specify a value he considers to be high, medium, low, etc.and we can calculate the degree of membership of each alternative in the fuzzy set. These values can also be subjectively assigned after examination of the attribute data. Ranges of values can be specified as we did for the decision to close a customer service center.

The rough sets formulation as the basis for determining the decision rules is easily performed through maximization and minimization of combinations of the fuzzy set values. The process is not computationally intensive, although it does become more labor intensive beyond the two attribute with one decision case presented in this paper. The authors hope to have a computer program available in the near future to handle large-scale problems.

Again, we stress that the proposed method does not give an answer to: "are the decisions made, good decisions?". It is assumed that the expert is knowledgeable about the conditions under which the decision will be made. Our methodology gives an answer to "how closely does the expert follow the attributes under consideration in making his decision?". If the decisions seem to closely follow the values of the attributes, then strong rules can be acquired through examples and the expert's knowledge can be put into machine representable form.

At this time, HL&P has not made a decision to close either of the customer service centers. Management has relied on reducing the operating costs at each of the centers by moving to the company's downtown Houston location, the CSC employees who generally had only telephone contact with district customers. A complete evaluation of the data from Tables 1, 2, and 3 is to be performed and submitted to HL&P as soon as the prototype computer program is completed.

REFERENCES

Arciszewski, T. and Ziarko, W. 1986. "Adaptive expert system for preliminary engineering design," Proceedings 6th International Workshop on Expert Systems and their Applications. Avignon, France, 1, 696-712 .

Bobrow, D.G., Mittal, S. and Stefik, M.J. 1986. "Expert systems perils and promises" Communications of the ACM,29, 880-894.

Fibak, J., Slowinski, K. and Slowinski, R. 1986. "The application of rough set theory to the verification of indications for treatment of duodenal ulcers by HSV," Proceedings 6th International Workshop on Expert Systems and their Applications, Avignon, France, 1, 587-594.

Grzymala-Busse, J.W. 1988. "Knowledge acquisition under uncertainty: a rough set approach," Journal of Intelligent and Robotic Systems, 1, 3-16.

Mamdani, A., Efstathiou, J. and Pang, D. 1985. "Inference under uncertain expert systems 85," Proceedings Fifth Technical Conference British Computer Society, Specialist Group on Expert Systems, 181-194.

Mrozek, A. "Information systems and control algorithms", 1985. Bulletin Polish Academy of Science,Technical Science,33,195-204.

Mrozek, A. 1987. "Rough sets and some aspects of expert systems realization," Proceedings 7th International Workshop on Expert Systems and their Applications, Avignon, France, 597-611.

Pawlak, Z. 1981. "Rough sets. Basic Notions," Institute Computer Science, Polish Academy of Science Report No. 431, Warsaw.

Pawlak, Z. 1981. "Classification of objects by means of attributes," Institute Computer Science Polish Academy Science Report No. 429, Warsaw.

Pawlak, Z. 1982. "Rough sets," International Journal of Information Computer Science,11, 341-356.

Pawlak, Z. 1983. "Rough classifications," International Journal of Man-Machine Studies, 20, 469-483.

Pawlak, Z. 1985. "Rough sets and fuzzy sets," Fuzzy Sets and Systems, 17, 99-102.

Wiederhold, G.C., Walker, M., Blum, R., and Downs, S. 1986."Acquisition of knowledge from data," Proceedings ACM SIGART International Symposium on Methodologies for Intelligent Systems, Knoxville, Tennessee, 78-84.

Zadeh, L.A. 1983. "The rule of fuzzy logic in the management of uncertainty in expert systems," Fuzzy Sets and Systems11,119-227.

Zadeh, L.A. 1979. "Fuzzy sets and information granularity,"Advances in Fuzzy Set Theory and Applications, 3-18.

Zadeh, L.A. 1981. "Possibility theory and soft data analysis," Mathematical Frontiers of the Social and Policy Sciences. Eds. L. Cobb and R.M. Thrall, 69-129. Westview Press, Boulder, Colorado.

N93
29562
UNCLAS

FUZZY SIMULATION IN CONCURRENT ENGINEERING

A.Kraslawski, L.Nyström

Department of Chemical Technology, Lappeenranta University of Technology,
Box 20, SF- 53851 Lappeenranta, Finland

ABSTRACT

Concurrent engineering starts to be more and more important practice in manufacturing.
One of the problems in concurrent engineering is uncertainty in the values of input variables as well as operating conditions.
The problem solved in the presented paper consists in the simulation of processes where the raw materials and the operational parameters with fuzzy characteristics are applied. The processing of fuzzy input information is performed by the vertex method and commercial simulation packages POLYMATH (1990) and GEMS(1987). The examples are presented to illustrate the usefulness of the method to the simulation of chemical engineering processes.

INTRODUCTION

There are two main reasons to model uncertain knowledge in chemical engineering. The first one is the scale of the phenomena. The micro scale is of the growing interest for the chemical engineers. The key examples are the biochemical processes and new materials technologies. The analysis on the level of agglomerates, cells or molecules is of the other type than that on meso or micro scale.
The second reason is the global change of the estimation of technologies in the surrounding world. Environmental, economic, and cultural analysis is needed now to answer the question: Is a given technology good or not ?
In both cases, the sources of the uncertainty is the process complexity. The more complex the process is, the less information could be presented in numerical and objective way. Such a situation is a consequence of the behaviour of complex systems. It is not a consequence of the lack of the good tools of analysis.
The chemical engineers have to realize that another type of processes requires another tools for understanding and description.

The change in design process is the additional reason for the use of fuzzy calculations.

Design in chemical engineering is a long and complex process.
Its consequences are a long product development cycle, high manufacturing costs, and

often, poor final quality. The main reason for this situation is the sequential nature of the design process. This way of proceeding results from the fact that the design objectives and constraints are formulated gradually in all stages of the product and process development. In the next step the product is tested and, if the criteria are not achieved, then the design procedure is repeated.
The name of such method is serial engineering.

The present chemical process industry situation is characterized by the growing competition, rising degree of complexity, and demand of high quality products. To survive in a new situation, the companies have to reduce the time from market demand to the full scale production to reduce the costs and to be more flexible. These demands evoke the need of new managerial as well as engineering techniques.

One of the new engineering methods is concurrent engineering (Rosenblatt and Watson, 1991, Ishii, 1990 and Hartley 1991).
Its essence is an integration of various manufacturing, marketing and engineering activities. This demand is realized by the team work of the multi-disciplinary groups. The people from marketing, design, manufacturing, sales, and services are working together. They formulate the required properties of a product, transform them into the engineering data, study the resulting manufacturing problems and establish final parameters of the product.

The tool for communication inside such a multidisciplinary team is a "house of quality", (Hauser and Clausing 1988, Thackeray and van Treeck 1990).

The integration of various activities results in the simultaneous generation and evaluation of the different variants of product and process. The comparison of both types of engineering is presented in Fig 1.

The imprecise values of raw materials properties, operation parameters as well as product demands are the consequences of the application of the concurrent engineering tools in chemical engineering problems.
The properties of raw materials are imprecise especially in batch and bio- processes. It is due to the nonhomogeneity of the substrates that is normal in noncontinuous processes as well as in natural products.

The problems of the imprecision in the operational parameters reflect the fact of the contradictory conditions imposed on the process. The contradictions are the consequences of the fulfilling of the different criteria. Given criterion could be reached at the given set of parameters. In order to obtain the reasonable solution the compromise has to be reached. The result of such compromise is the creation of the operational ranges for parameters instead of crisp values of variables.

The variability of demand is a popular situation that results from the market changes. Uncertainty of stochastic type could be treated by the well known probabilistic methods. However, there has been very few attempts to take directly into account the non-stochastic lack of precision in simulation as well as in optimization (Edgar and Himmelblau 1989). There are several approaches to study the influence of uncertainty on the output variables. The most popular methods are flexibility and sensitivity analysis. They are complicated and time consuming. The direct introduction of fuzzy variables into the existing packages is the simplest way to analyse the non-stochastic uncertainty. There are several approaches to study the uncertainty in design (Wood et al. 1991). In the present paper the vertex method is applied to introduce the fuzzy values into the existing programs.

VERTEX METHOD

The vertex method is based on the α - cut concept and the interval analysis proposed by Dong and Shah (1987). It enables the calculation of the membership function μ_y of the following expression:

$$y = f\,(x_1\,,...,\,x_n) \qquad (1)$$

where $x_1\,,...,\,x_n$ are fuzzy variables.
Let us assume the triangular form of μ_{x_i} membership function. At the given α- level, the values of the membership function are $[a_\alpha,\, b_\alpha]$ as it is shown in Fig. 2. As a result for the given $x_1,..., x_n$ and α - cut one obtains the set of intervals $[a_1,\, b_1]$, ..., $[a_n,\, b_n]$. The set of the intervals forms an n-dimensional region with 2^n vertices. An example for n=2 is given in Fig. 3. To obtain the y value in Eq.1 on the α - level one has to calculate:

$$y_1 = f\,(c_1)\,,...,\,y_r = f\,(c_r) \qquad (2)$$

where $c_1 = (a_1\,,...,\,a_n)\,,...,\,c_r = (b_1\,,...,\,b_n)$.
The y value in Eq. 1 at the level α is expressed as the interval function:

$$Y = [\min f\,(c_k)\ ,\ \max f\,(c_k)] \qquad (3)$$

The values of Y calculated on the different α - levels create the output fuzzy values as presented in Fig.3.
If the membership functions of fuzzy variables are triangular, then the number of runs equals 2^n , where n is a number of fuzzy variables. The fuzzy output is determined at different α - levels according to Eq.3. The Y values are calculated for $\alpha = 1$ and $\alpha = 0$, in this paper, for the sake of simplicity.

EXAMPLES

The examples presented below illustrate the fuzzy simulation in concurrent engineering problems. The fuzzy forms of operating conditions as well as raw material properties are obtained applying "the house of quality " method.
The imprecision of the operation conditions and physico-chemical properties is

studied in the first example.
The influences of imprecise parameters of raw materials and operational conditions are examined in the second example.

Example 1.

The following reactions has been studied by Himmelblau (1970):

$$A+B \xrightarrow{k_1} C+F$$

$$A+C \xrightarrow{k_2} D+F$$

$$A+D \xrightarrow{k_3} E+F$$

The proposed model for the reacting system is as follows:

$$\frac{dA}{dt} = - k_1AB - k_2AC - k_3AD$$

$$\frac{dB}{dt} = - k_1AB$$

$$\frac{dC}{dt} = k_1AB - k_2AC$$

$$\frac{dD}{dt} = k_2AC - k_3AD$$

$$\frac{dE}{dt} = k_3AD$$

where the kinetic constant $k_i = a_i$ exp(-' $_i$ / T) i=1,2,3.
a_i , b_i are constants and T is process temperature.
The initial and final conditions with concentration expressed in mole/liter and time in minutes has been reported as:

A(0) = 0.0209 , B(0) = 0.00697, C(0) = D(0) = 0 and t =200.

The nominal values of kinetic constants have been reported as:

$k_1 = 14.7, k_2 = 1.53, k_3 = 0.294.$

Aim of the study is to determir:ː the sensitivity of the concentrations A and E caused by the imprecise values T.
Temperature T is given in the form of fuzzy set. It is a consequence of the appplication of the "house of quality" method.

Solution

The following fuzzy kinetic constants result from the "house of quality " method:

$k_1 = (13.200 \quad 14.700 \quad 15.500)$
$k_2 = (\ 1.180 \quad 1.530 \quad 1.720)$
$k_3 = (\ 0.253 \quad 0.294 \quad 0.315)$

There are $n = 2^3 = 8$ vertices c_i according to vertex method. Given vertex c_i is a vector composed of three kinetic constants. The calculations should be realised at different α - levels of the fuzzy kinetic constants.
The values of all vertices at $\alpha = 0$ are presented in Table 1.
If $\alpha = 1$ then calculations are performed only in one vertex c ($k_1 = 14.7$, $k_2 = 1.53$, $k_3 = 0.294$).The system of differential equations has to be solved for all the combinations of the kinetic constants' values.

As a result the profiles of concentrations A and E are obtained using (POLYMATH 1990).
Because there are no extremal points, fuzzy values of A and E after time t = 200 min could be determined from the Eq.3. In the opposite situation instead of applying Eq.3. another approach should be used (Wood et al. 1991).
At $\alpha = 0$, according to Table 1, minimal and maximal values of A and E are:
min A = 0.00551 , max A = 0.00631, min E = 0. 00151, max E = 0.00189.
At $\alpha = 1$ there is only one point to calculate. The results of simulation are given in Table 1 for vertex 9.

The resulting fuzzy concentrations A and E obtained for fuzzy kinetic coefficients are as follows:

A = (0.00551 0.00573 0.00631)
E = (0.00151 0.00177 0.00189)

Example 2

The problem consists in the estimation of the product characteristics that are influenced by the the imprecise properties of raw material and operation conditions. The process under consideration is mechanical pulp mill peroxide bleaching. The raw material is unbleached pulp and the product is bleached pulp.
The raw material properties are light scatttering, brightness, and initial pulp pH. The

operating conditions are limited in this example to initial peroxide concentration. The product properties are brightness, final pH and metal ions content.
The process has been simulated with (GEMS 1987). Usually, the raw material is not uniform. As a result, its parameters could not be determined in a precise way. As a consequence, the operating parameters are uncertain, too.
The aim is to estabilish the product properties taking into account the imprecision of raw materials properties and operating conditions
Solution

The form of fuzzy input variables of raw material is presented in Table 2. The results of some simulations are presented in Table 3. Resulting interval values are given in Table 4. The fuzzy characteristic of products is determined by Cartesian product of two fuzzy sets y_1 and y_2 (Dubois and Prade 1988). The resulting fuzzy set is as follows:

$$\frac{0}{(7.3;\ 68.77)} + \frac{0}{(7.3;\ 71.85)} + \frac{0}{(8.53;\ 68.77)} + \frac{0}{(8.53;\ 71.85)} +$$

$$\frac{1}{(8.75;\ 75.84)} + \frac{1}{(8.93;\ 74.32)} + \frac{1}{(8.93;\ 75.64)}$$

CONCLUSIONS

The presented method can be used in concurrent engineering approach to process design. The main advantage of the presented method over the existing approaches is its ability to study the uncertainty in the raw materials characteristic as well as in operating conditions. It could be used with commercial packages without any changes of the existing programs. The construction of the " compact " package is the main aim for the future. Such a package should be composed of simulator, vertex method module and house of quality interactive program.

REFERENCES

Dong W. and H.C. Shah (1987). Vertex method for computing functions of fuzzy variables. Fuzzy Sets and Systems 24, 65-78.

Dubois D. and H. Prade (1988). Théorie des possibilités. Masson, Paris.

Edgar T.F. and D.M. Himmelblau (1989). Optimization of Chemical Process. McGraw-Hill

GEMSOP-GEMS Optimization Package (1987). Reference Manual. Applied Process Control Inc., Moscow, Idaho, USA

Hauser J.R. and D. Clausing (1988). The house of quality. Harvard Business Review, May- June, 63-73.

Hartley J. (1991). Simultaneous Engineering. Industrial Newletters, Beds.

Himmelblau D. M. (1970). Proces Analysis By Statistical Methods. Wiley & Sons, New York.

Ishii K. (1990). The role of computers in simultaneous engineering. In Computers in Engineering 1990, vol. 1, pp 217-224. ASME, New York.

POLYMATH (1990) User- Friendly Numerical Analysis Programs, Eds. M.B. Cutlip and M. Shacham. CACHE Corp., Austin.

Rosenblatt A. and G.F. Watson (1991). Concurrent Engineering. IEEE Spectrum , July 1991, pp 22-37.

Thackeray R. and G. van Treeck (1990). Applying quality Function deployment for software product development. J. Engng. Design 1, 389-410.

Wood K., K. Otto and E. Antonsson (1991). Engineering Design Calculations under Uncertainty. IFEC' 91. Yokohama .

Table 1. The vertices coordinates of kinetic constants and simulation results.

vertex number	k_1	k_2	k_3	A 10^2	C 10^2	D 10^2	E 10^2
1	13.2	1.18	0.253	0.631	0.854	0.460	0.151
2	13.2	1.18	0.315	0.606	0.881	0.431	0.178
3	13.2	1.72	0.253	0.577	0.415	0.495	0.161
4	13.2	1.72	0.315	0.551	0.436	0.464	0.189
5	15.5	1.18	0.253	0.630	0.850	0.460	0.151
6	15.5	1.18	0.315	0.606	0.881	0.431	0.178
7	15.5	1.72	0.253	0.577	0.415	0.495	0.161
8	15.5	1.72	0.315	0.551	0.436	0.464	0.189
9	14.7	1.53	0.294	0.573	0.544	0.466	0.177

Table 2. Charecteristics of raw material

x_1 light scattering	71.9; 72.2; 15.0; 0.3
x_2 brightnes	63.2; 63.8; 1.1; 0.0
x_3 pH	10.8; 11.0; 0.6; 0.0

Table3. Examples of the results of simulation

x_1 light scattering	x_2 brightness	x_3 pH	x_4 H_2O_2	y_1 pH	y_2 brightness	y_3 metal ions
71.90	63.20	10.80	3.0	8.75	74.32	$6.87*10^{3}$

Table 4. Fuzzy output parameters of pulp

y_1 pH	0 1	8.50; 8.53 7.52; 7.69
y_2 brightness	0 1	68.77;69.7 74.54;75.8
y_3 10^3 metal ions	0 1	5.12;6.48 5.49;5.50

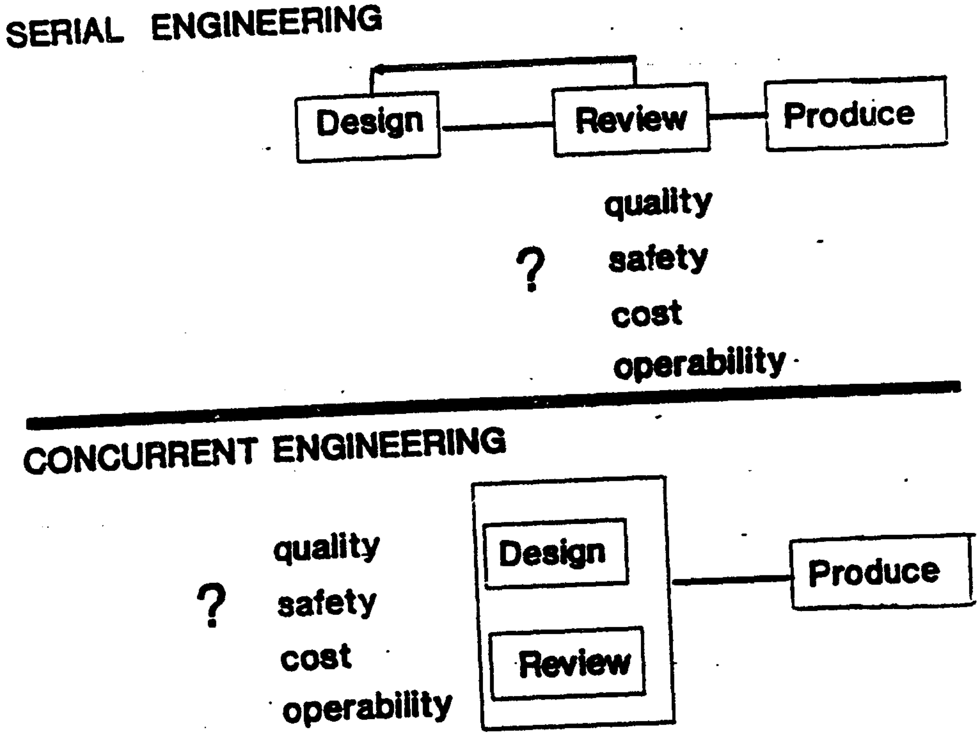


Fig. 1. Comparison of serial and concurrent engineering

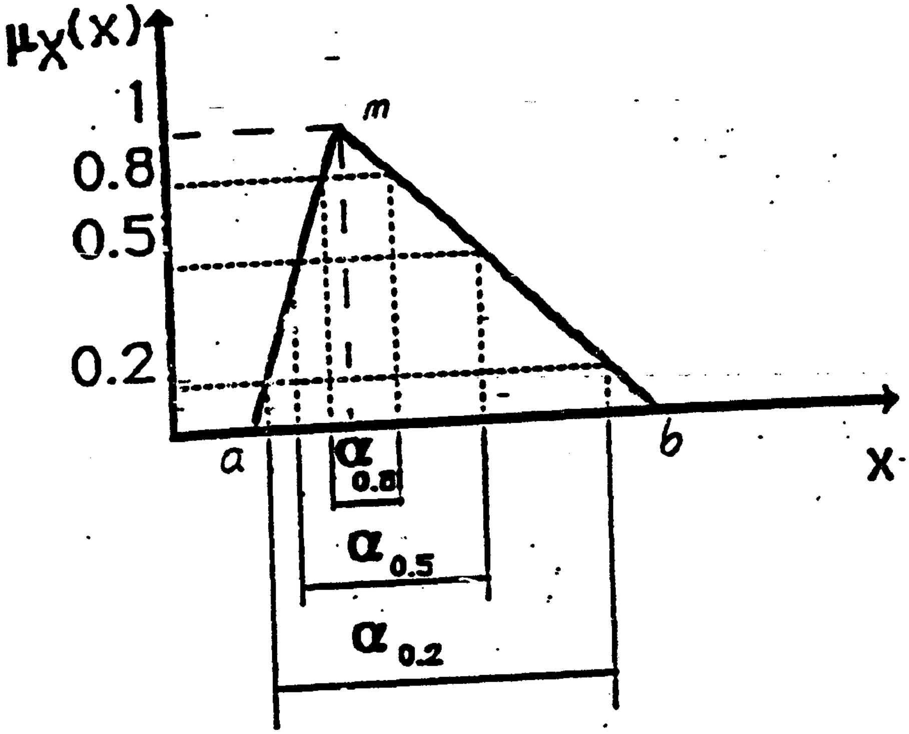
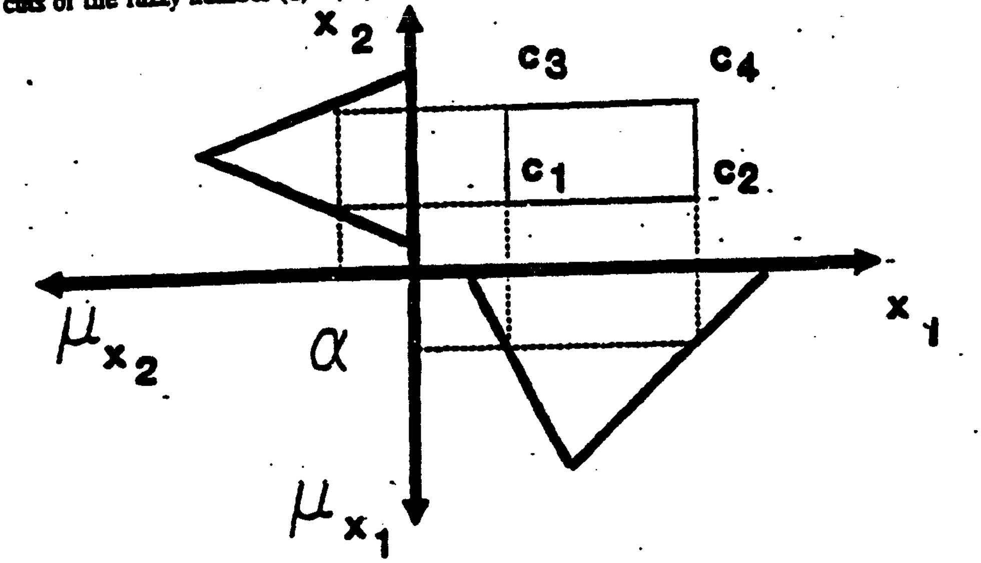


Fig. 2. α- cuts of the fuzzy number (a, m, b)



$$y_1 = f(c_1) \ \ldots \quad y_4 = f(c_4)$$

$$Y_\alpha = (\min f(c_k), \max f(c_k)) \qquad k = 1..4$$

Fig. 3. Vertex method

N93
29563
UNCLAS

INVERSE PROBLEMS: FUZZY REPRESENTATION OF UNCERTAINTY GENERATES A REGULARIZATION

V. Kreinovich[1], Ching-Chuang Chang[1], L. Reznik[2], G. N. Solopchenko[3]
[1]Computer Science Department, University of Texas at El Paso, El Paso, TX 79968 USA
[2]Department of Electrical and Electronic Engineering, Footscray Campus
Victoria University of Technology, MMC Melbourne, VIC 3000 Australia
[3]St. Petersburg Technical University, St. Petersburg 195251, Russia

Abstract. In many applied problems (geophysics, medicine, astronomy, etc) we cannot directly measure the values $x(t)$ of the desired physical quantity x in different moments of time, so we measure some related quantity $y(t)$, and then we try to reconstruct the desired values $x(\cdot)$. This problem is often *ill-posed* in the sense that two essentially different functions $x(t)$ are consistent with the same measurement results. So, in order to get a reasonable reconstruction, we must have some additional prior information about the desired function $x(t)$. Methods that use this information to choose $x(t)$ from the set of all possible solutions are called *regularization methods.*

In some cases, we know the statistical characteristics both of $x(t)$ and of the measurement errors, so we can apply statistical filtering methods (well-developed since the invention of a Wiener filter). In some situations, we know the properties of the desired process, e.g., we know that the derivative of $x(t)$ is limited by some number Δ, etc. In this case, we can apply standard regularization techniques (e.g., Tikhonov's regularization).

In many cases, however, we have only uncertain knowledge about the values of $x(t)$, about the rate with which the values of $x(t)$ can change, and about the measurement errors. In these cases, usually one of the existing regularization methods is applied. There exist several heuristics that choose such a method. The problem with these heuristics is that they often lead to choosing different methods, and these methods lead to different functions $x(t)$. Therefore, the results $x(t)$ of applying these heuristic methods are often unreliable.

We show that if we use fuzzy logic to describe this uncertainty, then we automatically arrive at a unique regularization method, whose parameters are uniquely determined by the experts knowledge. Although we start with the fuzzy description, but the resulting regularization turns out to be quite crisp.

1. INTRODUCTION

What is an inverse problem ([TA77], [I83], [G84], [I86], [I86a], [LRS86], [CB86]). In many applied problems (geophysics, medicine, astronomy, etc) we cannot directly measure the values $x(t)$ of the desired physical quantity x in different moments of time, so we measure some related quantity $y(t)$, and then try to reconstruct the desired values $x(t)$. For example, in case the dependency between $x(t)$ and $y(t)$ is linear, we arrive at a problem of reconstructing $x(t)$ from the equation $y(t) = \int k(t,s)x(s)\,ds + n(t)$, where $k(t,s)$ is an approximately known function, and $n(t)$ denote the (unknown) errors of measuring $y(t)$. These problems are called *inverse problems.*

Another example of inverse problems is *image reconstruction* from a noisy raw data.

Why inverse problems are so difficult to solve? These problems are often *ill-posed* in the sense that two essentially different functions $x(t)$ are consistent with the same observations $y(t)$. For example, since all the measurement devices are inertial and thus suppress the high frequencies, the functions $x(t)$ and $x(t) + sin(\omega t)$, where ω is sufficiently big, lead to almost similar values of

$y(t)$. So, in order to get meaningful results, we must somehow choose from all possible solutions $x(t)$ (i.e., from all the functions that are consistent with the measurement results) a one that is the most reasonable, the most *regular* (in some sense). A process of choosing such a function is therefore called a *regularization* [TA77], [I83], [G84], [I86], [I86a], [LRS86].

Inverse problems are extremely important for space exploration. If we are analyzing familiar processes, then we usually know (more or less) how the function $x(t)$ looks like. For example, we can know that $x(t)$ is a linear function $x(t) = C_1 + C_2 x$, or a sine function $x(t) = C_1 sin(C_2 t + C_3)$, etc. In mathematical terms, we know that $x(t) = f(t, C_1, ..., C_k)$, where f is a known expression, and the only problem is to determine the coefficients C_i. This is how, for example, the orbits of planets, satellites, comets, etc., are computed: the general shape of an orbit is known from Newton's theory, so we only have to estimate the parameters of a specific orbit. In such cases, the existence of several other functions $x(t)$ that are consistent with the same observations, is not a big problem, because we choose only the functions $x(t)$ that are expressed by the formula $f(t, C_1, ..., C_k)$.

In space exploration one of the main objectives (and the main challenges) is to analyze new phenomena, new effects, qualitatively new processes, and in these cases no prior expression f is known.

How these problems are traditionally solved? If we know the statistical characteristics of $x(t)$ and statistical characteristics of the measurement errors $n(t)$, then we can formulate the problem of choosing the maximally probable $x(t)$ and end up with one of the methods of *statistical regularization*, or *filtering* (Wiener filter is one of the examples of this approach).

If we do not have this statistical information, but we know, e.g., that the average rate of change of $x(t)$ is smaller than some constant Δ (i.e., $\sqrt{\int \dot{x}(t)^2 \, dt} \le \Delta$), then we can apply regularization methods proposed by A. N. Tikhonov and others [TA77], [G84], [LRS86].

In particular, one of the most widely used (and most efficient) regularization techniques consists of choosing among all the $x(t)$ that are consistent with given observations, a function $x(t)$ for which the so-called *Tikhonov functional* (or *Tikhonov stabilizer*)

$$J(\mu) = a_0 \int (x(t))^2 \, dt + a_1 \int (\dot{x}(t))^2 \, dt + a_2 \int (x^{(2)}(t))^2 \, dt + ... + a_k \int (x^{(k)})^2 \, dt$$

takes the smallest possible value, where a_i are non-negative real numbers, $a_k > 0$, $k \ge 1$, and $x^{(i)}(t)$ denotes $i-$th derivative of $x(t)$.

For *image reconstruction* problems, when instead of a function $x(t)$ of one variable t we have a function $I(x, y)$ of two coordinates (that expresses brightness in a point (x, y)), a similar functional that involves partial derivatives can be used.

If no such information is available, it is usually recommended to use Tikhonov's (or alternative) regularization techniques that correspond to some values of a_i. Several semi-heuristic rules of choosing these parameters a_i are known. The problem with these choices is that different rules sometimes lead to drastically different results, and therefore these results are unreliable.

Usually experts possess some uncertain knowledge. The whole situation seems hopeless, but it is not. Yes, in new fields we do not have precise knowledge of what is going on, but we may be able to make some uncertain predictions. For example, if we want to know how the temperature on a planet changes with time t, then the experts can tell that most likely, $x(t)$ is limited by some value M, and that the rate $\dot{x}(t)$ with which the temperature changes, is typically (or "most likely,", etc) limited by some value Δ, etc. We can also have some expert knowledge about the error, with which we measure $y(t)$, so the resulting expert's knowledge about the value of $y(t)$ in some point t looks like "the difference between the measured value $y(t)$ and the actual value $Y(t)$ is most likely, not bigger than δ" (where δ is a positive real number given by an expert).

The importance of this information is stressed in [B92].

What we are planning to do. In the present paper we show that if we use fuzzy logic to describe this uncertainty, then we automatically arrive at a unique regularization method, whose parameters are uniquely determined by the experts knowledge. Moreover, although we start with the fuzzy description, but the resulting regularization turns out to be quite crisp.

In Section 2 we will discuss briefly how to choose an appropriate representation of the experts uncertainty, in Section 3 we use the resulting representations to solve the inverse problems.

2. PRELIMINARY DISCUSSION: HOW TO DESCRIBE RELATED UNCERTAINTY

What we have to describe. We want to use fuzzy logic to describe this kind of uncertainty. So we must do the following:

- find appropriate fuzzy representations of the experts statements of the type "most likely, X is $\leq M$", or "most likely, $|X - a| \leq \delta$", where X is unknown, and M, a, δ are known values;
- choose a way to combine the resulting fuzzy statements into a membership function for different $x(t)$;
- transform this fuzzy description of $x(t)$ into a single function $x(t)$ that will be produced as a solution of the inverse problem, i.e., choose an appropriate *defuzzification*.

In the present Section we will describe how to make all three choices. Actually we will start with choosing an appropriate combination rule, then we will choose an appropriate membership function, and then it will turn out that defuzzification is trivial.

How to choose an aggregation function. In general, our uncertain knowledge about the unknown function $x(t)$ consists of the statements of the following types: "most likely, $|x(t)| \leq M$", "most likely, $|\dot{x}(t)| \leq \Delta$", "most likely, $|y(t) - \int k(t,s)x(s)\,ds| \leq \delta$", etc. Each statement is fuzzy in the sense that for an arbitrary function $x(t)$ we are not 100% sure whether this statement is true for this function or not. The general idea of fuzzy logic is to describe this uncertainty by a *membership function*, i.e., by a mapping that assigns to every $x(t)$ a number from the interval [0,1], that describes to what extent we believe that this statement is true.

Suppose that we have already decided how to express each of previous statements in terms of membership functions. So we get a different membership function for each moment of time t and for each statement. We must now generate a membership function that describes all our knowledge, i.e., that describes the fact that the first statement is true, *and* the second statement is true, etc. The total knowledge is obtained by applying "and" to all the statements, and therefore the resulting membership function must be obtained by applying one of the operations $\& : [0,1] \times [0,1] \to [0,1]$ that express "and" to all the correspondent membership functions $\mu_i(t)$: $\mu(x(t)) = \mu_1(x(t))\&\mu_2(x(t))\&\ldots$

Experimental results given in [HC76], [O77], and [Z78], show that among all possible "and"-operations $a, b \to min(a,b)$ and $a, b \to ab$ are the best fit for human reasoning. The *min* operation does not seem to be adequate for our purposes, because if we use *min*, then, e.g., the degree, to which a function $x(t)$ satisfies the condition "most likely, $|x(t)| \leq M$", is equal to the minimal of the degrees of the corresponding statements. This minimum is attained when the value of $|x(t)|$ is the biggest possible. Therefore, the function $x_1(t)$ that is everywhere equal to $2M$, gets the same degree of consistency with the above-given rule, as the function $x_2(t)$ that is almost everywhere equal to 0, and is attaining the value $2M$ only on a small interval. Intuitively, however, for the first function $x_1(t)$ (for which the inequality is always false), our degree of belief that it satisfies this condition is practically 0, while for the second function $x_2(t)$, for which this inequality is almost everywhere true, our degree of belief must be close to 1. So using *min* in our problem is inconsistent with our intuition, and therefore we must use the product for &.

Comment. Other arguments for choosing different & operations are given in [K83], [KR86], [K87], [KKM88], [K89], [K89a], [K90], [KK90], [KL90], [KQL91], [KQLFLKBR92].

What membership functions to choose? We want to describe the statements of the type "most likely, $|X - a| \leq \delta$", where X is an unknown ($x(t), \dot{x}(t)$, or $y(t)$) and a, δ are known values (for example, $\delta = M$ and $a = 0$). So we must describe, to what extent any given value x satisfies this condition.

Evidently, x satisfies the inequality $|x - a| \leq \delta$ if and only if the value $y = (x - a)/\delta$ satisfies the inequality $|y| \leq 1$. Therefore, it is natural to assume that the statement "most likely, $|x - a| \leq \delta$" has the same degree of belief as the statement "most likely, $|y| \leq 1$", where $y = (x - a)/\delta$. So, if we will be able to describe a membership function $\mu(y)$ that corresponds to the statement "most likely, $|y| \leq 1$", then we will be able to describe our degree of belief $\mu_1(x)$ that x satisfies the condition "most likely, $|X - a| \leq \delta$" as $\mu((x - a)\delta)$. So the main problem is to find an appropriate function $\mu(x)$.

In the present paper we use Gaussian membership functions $\mu(x) = exp(-\beta x^2)$ for some $\beta > 0$. Therefore, the statement "most likely, $|X - a| \leq \delta$" will be described by a membership function $\mu_1(x) = exp(-\beta(x - a)^2/\delta^2)$.

Gaussian membership functions are widely used in fuzzy systems and fuzzy control (see, e.g., [K75], [BCDMMM85], [YIS85], [KM87, Ch. 5], etc.), and there are several theoretical explanations why they are so successful: in [KR86] and in Section 8 of [KQLFLKBR92] we prove that Gaussian functions are optimal (in some reasonable sense), and in [KQR92] we describe reasonable axioms that uniquely determine Gaussian membership functions.

A remark about defuzzification. Suppose that we have determined the membership functions $\mu_i(x(t))$, that correspond to different statements about the unknown process $x(t)$. Then the resulting membership function $\mu(x(t))$ is obtained by multiplying the functions $\mu_i(x(t))$ that correspond to these statements.

All the values of μ_i are ≤ 1. So, if we multiply many such values, we end up with very small numbers. E.g., if we have 10 experts who all assign the truth value 0.9 to some event, the resulting estimate is $0.9^{10} \approx 0.3$. Thus, the fact that for some process $x(t)$ the membership value $\mu(x(t))$ is small, does not necessarily mean that this particular dependency $x(t)$ is hardly possible. What is meaningful is not the *absolute*, but the *relative* value of $\mu(x(t))$: if $\mu(x(t)) \ll \mu(y(t))$, then it does mean that, according or our knowledge, $x(t)$ is much less probable than $y(t)$.

To make these comparisons easier, L. Zadeh proposed to use *normalization*, i.e., turn from $\mu(x(t))$ to $\mu'(x(t)) = N\mu(x(t))$, where a normalizer N is chosen in such a way that the maximal value of $\mu'(x(t))$ is equal to 1 (i.e., $N = 1/(max\,\mu(x(t)))$).

Comment. Theoretical explanations of this choice of a normalization are given in [KQLFLKBR92] (in the framework of a general mathematical foundation scheme for fuzzy logic).

3. FUZZY DESCRIPTION OF RELEVANT EXPERTS KNOWLEDGE AND RESULTING REGULARIZATION

Let's first list the possible experts statements.

1) Usually experts can give the approximate range of the process $x(t)$, i.e., they can give a number M for which "most likely, for every t the value of $|x(t)|$ is limited by M."

2) Usually they can also give some approximate bounds for the rate, with which the values of $x(t)$ can change, i.e., they can give a number Δ, for which "most likely, for every t, the value of $|\dot{x}(t)|$ is limited by Δ".

3) Sometimes, the experts' knowledge and/or intuition can also prompt the approximate bounds for the second time derivative of the process (acceleration), and bounds for some higher derivatives. For each of these derivatives, an expert gives a value Δ_i and states that "most likely, for every t, the value of $|x^{(i)}(t)|$ is limited by Δ_i" (here $x^{(i)}(t)$ denoted i–th derivative).

4) Experts can also give some information about the possible measurement errors, i.e., about the values $n(t) = y(t) - \int k(t,s)x(s)\,ds$, where $y(t)$ are the measured values. In this case, an expert gives a value δ, and states that "most likely, for every t, the value of $|n(t)|$ is limited by δ."

In addition to that, we have some measurement results $y(t)$, and these measurement results determine a set X of all the functions that are consistent with them. For example, if we know the maximal possible value ε of a measurement error $n(t)$, then X consists of all the functions $x(t)$ that satisfy the inequality $|y(t) - \int k(t,s)x(s)\,ds| \leq \varepsilon$ for all t.

We want to represent the expert knowledge in terms of a membership function that is defined on this set X.

We cannot directly translate these statements into membership functions, so we need an additional approximation process. Each of these statements refers not to a single value of some variable, but to infinitely many values, namely, to the values of $x(t)$ for all possible moments of time t. So, if we write down all the resulting elementary statements, we will end up with infinitely many such statements. So, to get a membership function that coresponds to the resulting knowledge, we must apply an "and"-operator to infinitely many membership functions, that correspond to infinitely many elementary statements. But we know only how to apply "and"-operator to finitely many functions.

In order to cover the infinite case, we will apply the usual mathematical method of dealing with infinities: we will first consider the case, when the experts statements are applicable only to finitely many points $t_1, ..., t_n$, and then tend n to infinity in such a way that in the limit these points t_i are everywhere dense. One of the natural possibilities to do that is to choose $t_i = t_0 + ih$, where $h > 0$, and then take $t_0 \to -\infty$, $h \to 0$, and $n \to \infty$ in such a way that $t_n = t_0 + nh \to +\infty$.

The resulting membership function: derivation. Let us apply this procedure and compute the resulting membership function. The readers who are interested only in the final result can skip this subsection.

Let's first consider the case, when the only experts knowledge consists of the bounds M and Δ on $|x(t)|$ and $|\dot{x}(t)|$. Then for each t the corresponding membership functions are $exp(-\beta|x(t)|^2/M^2)$ and $exp(-\beta|\dot{x}(t)|^2/\Delta^2)$. Therefore, if we take into consideration these statements for $t = t_1, ..., t_n$, $t_i = t_0 + ih$, the resulting membership function will be equal to the product of these membership functions, i.e., will be equal to the following expression

$$\mu(x(t)) = \prod_{i=1}^{n} exp(-\beta|x(t_i)|^2/M^2) \times \prod_{i=1}^{n} exp(-\beta|\dot{x}(t_i)|^2/\Delta^2).$$

Comment. We are restricted to the set X of all functions $x(t)$ that are consistent with the measurement results. Therefore, the above xpression for $\mu(x(t))$ is valid only for such functions $x(t)$. All functions $x(t)$ that are not consistent with the measurement results are impossible, i.e., if $x(t) \not\in X$, then $\mu(x(t)) = 0$.

Since $exp(a) \times \exp(b) = exp(a+b)$, we can simplify the expression for $\mu(x(t))$ as follows:

$$\mu(x(t)) = exp(-(\beta/M^2)\sum_{i=1}^{n}|x(t_i)|^2 - (\beta/\Delta^2)\sum_{i=1}^{n}|\dot{x}(t_i)|^2).$$

What happens when $n \rightarrow \infty$? If we multiply the sum $\sum_{i=1}^{n}|x(t_i)|^2$ by $h = t_{i+1} - t_i$, we get an integral sum for the integral $\int |x(t)|^2\,dt$. These integral sums tend to this integral, when $h \rightarrow 0$. Hence, for small h, this sum is approximately equal to $h^{-1}\int|x(t)|^2\,dt$. Therefore, the membership function is approximately equal to the following expression:

$$\mu_h(x(t)) \approx exp(-(\beta/h)J(x(t))),$$

where

$$J(x(t)) = M^{-2}\int|x(t)|^2\,dt + \Delta^{-2}\int|\dot{x}(t)|^2\,dt.$$

When $h \rightarrow 0$, $(\beta/h)J(x(t)) \rightarrow \infty$, and, therefore, $\mu_h(x(t)) \approx exp(-(\beta/h)J(x(t))) \rightarrow 0$. Therefore, if we apply a transition to a limit, we end up with a meaningless expression $\mu(x(t)) \equiv 0$.

In order to get a reasonable limit membership function $\mu(x(t))$, we must apply the normalization procedure before going to a limit. In other words, we must transform $\mu_h(x(t))$ into $\mu'_h(x(t)) = N\mu_h(x(t))$, where $N = 1/(max_{x(t)\in X}\mu_h(x(t)))$.

Since $\mu_h(x(t)) \approx exp(-(\beta/h)J(x(t)))$, the value of $\mu_h(x(t))$ is the biggest when the value of $J(x(t))$ is the smallest possible. So, if we denote by m the smallest possible value of the functional $J(x(t))$ on X, we can conclude that $max_{x(t)\in X}\mu_h(x(t)) = exp(-(\beta/h)m)$. Therefore, $N = 1/max = exp((\beta/h)m)$, and $\mu'_h(x(t)) = N\mu_h(x(t)) = exp(-(\beta/h)(J(x(t)) - m))$.

Now we are ready to describe the membership function $\mu(x(t)) = lim_{h\rightarrow 0}\mu'_h(x(t))$ that corresponds to the limit $h \rightarrow 0$. If $J(x(t)) = m$, then $\mu'_h(x(t)) = 1$, and therefore $\mu(x(t)) = 1$. If $x(t) \in X$ and $J(x(t)) \neq m$, then, since m is a minimum of $J(x(t))$, we get $J(x(t)) > m$, therefore $(\beta/h)(J(x(t)) - m) \rightarrow \infty$, and hence, $\mu'_h(x(t)) \rightarrow 0$ as $h \rightarrow 0$.

As a result, we get a crisp membership function that corresponds to Tikhonov's regularization. If $J(x(t)) \neq m$, we have $\mu(x(t)) = 0$. So, although we started with *fuzzy* statements and fuzzy membership functions, the resulting membership function is *crisp*: it is equal either to 1 or to 0 depending on whether the functional $J(x(t))$ attains its minimum at $x(t)$ or not. Hence, in this case, we do not need any defuzzification procedure: we just pick a function $x(t)$ from X, for which $J(x(t))$ attains its minimal value.

What if the experts can also give some bounds on the second and higher derivatives of the process $x(t)$. In case an expert gives estimates Δ_i for i-th derivative and/or a bound δ for the measurement error, the resulting membership function is the same, with the only difference that additional terms are added to $J(x(t))$: $\Delta_i^{-2}\int(x^{(i)})^2\,dt$ in case of i-th derivative, and

$$\delta^{-2}\int(y(t) - \int k(t,s)x(s)\,ds)^2\,dt$$

in case of an error bound.

How to solve inverse problems: resulting procedure. As a result, we arrive at the following methods of solving inverse problems:

1) ask an expert to give approximate bounds M for $|x(t)|$ and Δ for $|\dot{x}(t)|$; if possible, get also his bounds Δ_i for i–th derivative $|x^{(i)}(t)|$, and δ for the measurement error $|y(t) - \int k(t,s)x(s)\,ds|$;

2) from all the functions that are consistent with the measurement results, choose a function $x(t)$ for which the functional $J(x(t))$ attains the smallest possible value. In case the expert gives only the estimates M and Δ, $J(x(t)) = J_0(x(t)) + J_1(x(t))$, where $J_0(x(t)) = M^{-2} \int |x(t)|^2\,dt$ and $J_1(x(t)) = \Delta^{-2} \int |\dot{x}(t)|^2\,dt$. In case he gives bounds for i–th derivative and/or for errors, we must take $J(x(t)) = \sum_i J_i(x(t)) + J_e(x(t))$, where for $i > 1$ $J_i(x(t)) = \Delta_i^{-2} \int (x^{(i)}(t))^2\,dt$ and $J_e(x(t)) = \delta^{-2} \int (y(t) - \int k(t,s)x(s)\,ds)^2\,dt$.

We can use ready-made software. The resulting method turns out to be a particular case of the Tikhonov's regularization scheme. Therefore we do not need to design any new software: we can use the techniques, algorithms, and programs, that have already been developed for Tikhonov's regularization.

If the only thing we have done is justification of a well-known method, then what's the buzz? Our proposal to use Tikhonov's method has two advantages over the usual heuristic suggestion to use it:

i) Tikhonov's method is semi-heuristic, while we derived our method from the fuzzy formalism;

ii) we do not need any heuristic rule of choosing a_i, because we have explicit expressions for these parameters in terms of experts' bounds.

Therefore, we avoid the problem of Tikhonov's regularization that different heuristic rules lead to different values of a_i and, therefore, to different solutions $x(t)$.

4. CONCLUSIONS

Suppose that we must reconstruct $x(t)$ from the measurement results $y(t)$, and the problem is *ill-posed* in the sense that drastically different functions $x(t)$ are consistent with the same measurement results. Such problems are very frequent in geophysics, astronomy, image processing, etc. Suppose also that the only additional information that we have about the process $x(t)$ is the experts estimates M and Δ for which the experts say that "most likely, for every t the value of $|x(t)|$ is limited by M," and "most likely, for every t, the value of $|\dot{x}(t)|$ is limited by Δ", where $\dot{x}(t)$ denotes the rate with which $x(t)$ changes (i.e., in mathematical terms, time derivative of $x(t)$).

Then fuzzy representation of this uncertainty leads to the following method of using this experts' knowledge: from all the functions that are consistent with the measurement results, we choose a function $x(t)$, for which the functional $J(x(t))$ takes the minimal possible value, where $J(x(t)) = M^{-2} \int |x(t)|^2\,dt + \Delta^{-2} \int |\dot{x}(t)|^2\,dt$.

Similar functionals can be described for the cases, when bounds for higher derivatives and/or measurement errors are known.

The resulting method turns out to coincide with a particular case of the general Tikhonov's regularization approach. This approach has already been implemented in software, and it has been successfully tested on numerous real-life ill-posed problems.

The advantage of our approach is that we solve two main problems of Tikhonov's regularization:

- we provide a justification its formulas, and
- we provide a method for choosing the parameters of Tikhonov's regularization.

ACKNOWLEDGEMENTS

This work was supported by a NSF Grant No. CDA-9015006, NASA Research Grant No. 9-482 and the Institute for Manufacturing and Materials Management grant. The authors are thankful to M. Baker, C. Ferregut, and S. Nazarian (El Paso), and also to all participants of the 1992 International Fuzzy Systems and Intelligent Control Conference (Louisville, Kentucky), especially to L. Zadeh, S. Smith, and J. M. Barone, for valuable discussions.

REFERENCES

[B92] J. M. Barone *Fuzzy least squares and fuzzy entropy.* Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference, Louisville, KY, 1992, pp. 170–181.

[BCDMMM85] G. Bartolini, G. Casalino, F. Davoli, M. Mastretta, R. Minciardi, and E. Morten. *Development of perfomance adaptive fuzzy controllers with applications to continuous casting plants,* in: M. Sugeno (editor). *Industrial applications of fuzzy control,* North Holland, Amsterdam, 1985, pp. 73–86.

[CB86] I. Craig and J. Brown. *Inverse problems in astronomy.* Adam Hilget Ltd., Bristol, 1986.

[G84] V. B. Glasko. *Inverse problems of mathematical physics.* American Institute of Physics, N. Y., 1984.

[HC76] H. M. Hersch and A. Caramazza. *A fuzzy-set approach to modifiers and vagueness in natural languages.* J. Exp. Psychol.: General, 1976, Vol. 105, pp. 254–276.

[I83] *Inverse problems.* SIAM–AMS Proceedings, Vol. 14. American Mathematical Society, Providence, RI, 1983.

[I86] *Inverse problems.* Birkhauser Verlag, Basel, 1986.

[I86a] *Inverse problems.* Lecture Notes in Mathematics, Vol. 1225, Springer-Verlag, Berlin–Heidelberg, 1986.

[K75] A. Kauffman. *Introduction to the theory of fuzzy subsets. Vol. 1. Fundamental theoretical elements,* Academic Press, N.Y., 1975.

[KKM88] V. Kozlenko, V. Kreinovich, and M. G. Mirimanishvili. *The optimal method of describing the expert information.* In: Applied problems of systems analysis. Proceedings Georgian Polytechnic Institute, 1988, No. 8, pp. 64–67 (in Russian).

[K83] V. Kreinovich. *Foundations of the Maslov's operator.* In: Proceedings of the 3-rd National Conference on the Applications of Mathematical Logic, Tallinn, 1983, pp. 80–81 (in Russian).

[K87] V. Kreinovich. *Semantics of Maslov's iterative method.* In: Problems of Cybernetics. Vol. 131, Moscow 1987, pp. 30–63 (in Russian).

[K89] V. Kreinovich. *The optimal choice of formulas of fuzzy logic.* Center for New Informational Technology "Informatika", Leningrad, Technical Report, 1989 (in Russian).

[K89a] V. Kreinovich. *Optimization in case of uncertain optimality criteria: group-theoretic approach.* Center for New Informational Technology "Informatika", Leningrad, Technical Report, 1989 (in Russian).

[K90] V. Kreinovich. *Group-theoretic approach to intractable problems.* Lecture Notes in Computer Science, Springer, Berlin, Vol. 417, 1990, pp. 112–121.

[KK90] V. Kreinovich and S. Kumar. *Optimal choice of &- and ∨- operations for expert values* in: Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop, Fredericton, N.B., Canada, 1990, pp. 169–178.

[KL90] V. Kreinovich and A. M. Lokshin. *On the foundations of fuzzy formalism: explaining formulas for union, intersection, negation and modifiers.* University of Texas at El Paso, Computer Science Dept. Technical Report UTEP-CS-90-28, October 1990.

[KQL91] V. Kreinovich, C. Quintana, and R. Lea. *What procedure to choose while designing a fuzzy control? Towards mathematical foundations of fuzzy control,* Working Notes of the 1st International Workshop on Industrial Applications of Fuzzy Control and Intelligent Systems, College Station, TX, 1991, pp. 123–130.

[KR86] V. Kreinovich and L. K. Reznik. *Methods and models of formalizing prior information (on the example of processing measurements results).* In: *Analysis and formalization of computer experiments,* Proceedings of Mendeleev Metrology Institute, Leningrad, 1986, pp. 37-41 (in Russian).

[KQLFLKBR92] V. Kreinovich, C. Quintana, R. Lea, O. Fuentes, A. Lokshin, S. Kumar, I. Boricheva, and L. Reznik. *What non- linearity to choose? Mathematical foundations of fuzzy control.* Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference, Louisville, KY, 1992, pp. 349–412.

[KQR92] V. Kreinovich, C. Quintana, and L. Reznik. *Gaussian membership functions are most adequate in representing uncertainty in measurements.* University of Texas at El Paso, Computer Science Department, Technical Report, March 1992.

[KM87] R. Kruse and K. D. Meyer. *Statistics with vague data.* D. Reidel, Dordrecht, 1987.

[LRS86] M. M. Lavrentiev, V. G. Romanov, and S. P. Shishatskii. *Ill-posed problems of mathematical physics and analysis.* American Mathematical Society, Providence, RI, 1986.

[O77] G. C. Oden. *Integration of fuzzy logical information,* Journal of Experimental Psychology: Human Perception Perform., 1977, Vol. 3, No. 4, pp. 565–575.

[KR86] V. Kreinovich and L. K. Reznik. *Methods and models of formalizing prior information (on the example of processing measurements results).* In: *Analysis and formalization of computer experiments,* Proceedings of Mendeleev Metrology Institute, Leningrad, 1986, pp. 37-41 (in Russian).

[TA77] A. N. Tikhonov, V. Y. Arsenin. *Solutions of ill-posed problems.* V. H. Winston & Sons, Wasnington, DC, 1977.

[YIS85] O. Yagishita, O. Itoh, and M. Sugeno. *Application of fuzzy reasoning to the water purification process,* in: M. Sugeno (editor). *Industrial applications of fuzzy control,* North Holland, Amsterdam, 1985, pp. 19–40.

[Z65] L. Zadeh. *Fuzzy sets.* Information and control, 1965, Vol. 8, pp. 338–353.

[Z78] H. J. Zimmermann. *Results of empirical studies in fuzzy set theory.* In: *Applied General System Research* (G. J. Klir, ed.) Plenum, New York, 1978, pp. 303–312.

N93

29564

UNCLAS

S8-53
161848

P 10

N93-29564 5

QUANTIFICATION OF HUMAN RESPONSES

R.C. Steinlage*, T.E. Gantner*, P.Y.W. Lim**

Introduction. Human perception is a complex phenomenon which is difficult to quantify with instruments. For this reason, panels of several or many people are often used to elicit and aggregate subjective judgments. Print quality, taste, smell, sound quality of a stereo system, softness, and grading Olympic divers and skaters are some examples of situations where subjective measurements or judgments are paramount. We usually express what is in our mind through language as a medium but languages are limited in available choices of vocabularies, and as a result our verbalizations are only approximate expressions of what we really have in mind. For lack of better methods to quantify subjective judgments, it is customary to set up a numerical scale such as 1, 2, 3, 4, 5 or 1, 2, 3, ... , 9, 10 for characterizing human responses and subjective judgments with no valid justification except that these scales are easy to understand and convenient to use. But these numerical scales are arbitrary simplifications of the complex human mind; the human mind is not restricted to such simple numerical variations. In fact, human responses and subjective judgments are psychophysical phenomena that are fuzzy entities and therefore difficult to handle by conventional mathematics and probability theory. The fuzzy mathematical approach provides a more realistic insight into understanding and quantifying human responses. This paper presents a method for quantifying human responses and subjective judgments without assuming a pattern of linear or numerical variation for human responses. In particular, quantification and evaluation of linguistic judgments was investigated.

Approach. The method used to code responses obtained from panelists is especially important when one wishes to make decisions concerning properties or events which are not objectively quantifiable but which must be evaluated subjectively. The problem of coding such responses has been addressed from many directions. In this paper we propose a technique, based in fuzzy mathematics, for quantifying and evaluating subjective responses and then we test our technique in situations where the properties are also objectively measurable. By testing our technique in objective situations, we hope to lend credibility to its use in purely subjective situations. The technique we describe is a refinement of techniques originally proposed by Saaty [4-8].

Saaty [4-8] proposes using five adjectives as "response words" in subjective panel tests. These words indicate that two samples are indistinguishable with respect to a given property or that the difference between them is slight, moderate, significant, or extreme (or absolute). Of course, panelists are permitted to hedge their bets and cast their ballots between two such judgments.

Thus Saaty is proposing a 9 point scale for linguistic or subjective judgments as illustrated below. The illustration is stated in terms of physical weight although the particular property is irrelevant.

1 - A and B are equally heavy
2 -
3 - A is slightly heavier than B
4 -
5 - A is moderately heavier than B
6 -
7 - A is significantly heavier than B
8 -
9 - A is extremely heavier than B

The integers 2, 4, 6, 8 represent compromise judgments between two of the above odd numbered positions. As Saaty states, this is a good scale in that it provides enough shades of meaning without expecting a panelist to be scrupulous.

After obtaining panel data, the next problem is the analysis of this data. Aside from the usual statistical analysis, a technique that has been shown to be successful in fuzzy or subjective situations is to find the dominant eigenvalue and associated eigenvector for the reciprocal matrix of paired comparisons. This analysis is based on the work of Perron and Froebenius [1]. If n objects $A_1, \ldots, A_n$ are being compared, these are listed horizontally and vertically to indicate the rows and columns of a matrix M. If A_i is judged to be significantly heavier than A_j, then a 7 is placed in row i, column j and 1/7 is placed in row j, column i :

$$\begin{array}{c|ccccccc|}
 & 1 & \cdots & i & \cdots & j & \cdots & n \\
\hline
1 & & & & & \vdots & & \\
\vdots & & & & & & & \\
i & & & \cdots & & 7 & \cdots & \\
\vdots & & & & & & & \\
j & & \cdots & 1/7 & & & \cdots & \\
\vdots & & & \vdots & & & & \\
n & & & & & & & \\
\hline
\end{array}$$

If our objective is to determine the respective weights of n objects, then the resulting eigenvector should indicate the relative weights. If we have perfect information (no judgments are necessary and responses are not restricted to integers and their reciprocals) we could simply fill in the matrix using the ratio of the respective weights: $m_{ij} = w_i/w_j$. We then obtain a reciprocal matrix: $m_{ji} = w_j/w_i = 1/m_{ij}$.

$$M = \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{bmatrix}$$

It can be shown that $\lambda = n$ is the only non-zero eigenvalue for M and that $W = (w_1, \ldots, w_n)$ is its associated eigenvector; the correct weight determination is indeed obtained as the eigenvector. This eigenvector is unique up to a scalar multiple.

If the experiment was needed, however, perfect information is not available at the outset. But if the responses are a reasonable approximation to the reality of the situation, then the responses will approximate those which would have been placed in the "perfect information" matrix. Hence the eigenvalue should approximate n (the number of samples) and the associated eigenvector should approximate the actual distribution of the property (weight, etc.) among the samples. Thus the eigenvector not only ranks the samples ordinally (indicates smallest, largest, etc.) but also gives a cardinal ranking (indicates relative strengths or weights, etc.). In actuality $\lambda \geq n$, the associated eigenvector $V = (v_1, \ldots, v_n)$ is unique up to a multiplicative constant, and when normalized so that $v_1 + \ldots + v_n = 1$, v_i indicates the percentage of the total (weight) possessed by object i. The eigenvalue λ is a measure of the consistency of the responses given by the panelist. A good rule of thumb is that if $\lambda > n + 2$, the panelist has contradicted himself or herself so many times and/or so egregiously that his or her responses should be ignored. On the other hand, if λ is very close to n, the panelist was very consistent (although not necessarily accurate or correct). In short, the eigenvalue is a good flag to indicate errors in recording data; e.g., a number and its reciprocal may be interchanged, The 1 to 9 scale does conform well to linguistic comparisons in the sense that it allows one to discriminate simultaneously on $9 = 7 + 2$ levels. This is the maximum number in the range 7 ± 2 of simultaneous comparisons that an individual can keep in mind without becoming confused; see Miller [3]. If a scale much larger than 9 is used, the differences in reciprocals become negligible and some discrimination between samples in the resulting eigenvector will be lost. A collection of objects in which the samples may be too widely diverse should be subjected to a hierarchical analysis [4-8].

However, our experience indicates that while the above 1 - 9 scale may be appropriate for eliciting and coding human responses, it is not always the proper scale to be used in the ensuing matrix analysis. In fact, the scale used will be reflected in the results. The largest number used is in essence the ratio between the strongest and weakest (or heaviest and lightest, etc.) objects in the resulting eigenvector. Thus an inappropriate numerical scale will lead to undesirable end effects concerning the extremes of the objects being compared. This end effect is extremely volatile when computing percent error on the low end. Our experience indicates that a linear rescaling of the 1 - 9 linguistic scale to a scale determined by the accepted or perceived ratio of the two extreme objects in the given group significantly reduces this end-point effect.

__An Example:__ Consider 6 weights $w_1 = 2$, $w_2 = 4$, $w_3 = 6$, $w_4 = 8$, $w_5 = 10$, and $w_6 = 12$. The matrix $M_6 = [w_i/w_j] = [m_{ij}]$ is the matrix of perfect information, and the integer entries of this matrix range from 1 to 6. In this case, the weight ratios $w_2/w_1 = w_4/w_2 = w_6/w_3 = 2$ all indicate that the numerator weight is twice that of the denominator, which is quite different from the linguistic use of the number 2 in the above $1-9$ scale. The linguistic 2 says two samples are almost indistinguishable. The dominant eigenvalue of M_6 is 6 and its unit eigenvector is $V_6 = (w_1/w, \ldots, w_6/w)$, where $w = w_1 + \ldots + w_6 = 42$. We linearly rescaled the integer entries in M_6 to a $1-9$ scale to get a reciprocal matrix M_9, as well as to a $1-3$ scale to get a reciprocal matrix M_3. The unit eigenvectors V_9 and V_3, respectively, corresponding to the dominant eigenvalues of M_9 and M_3 generate weight vectors $42V_9$ and $42V_3$. These are displayed in the table below. In all cases the eigenvalue λ was less than 6.005. These low eigenvalues merely indicate consistency, not agreement with experimental measurements.

PERFECT INFORMATION
VARIOUS SCALES

	$42V_3$		$42V_6$		$42V_9$	
		% error		actual		% error
w_1	3.570	78.5%	2		1.344	-32.8%
w_2	5.418	35.5%	4		3.192	-20.2%
w_3	6.720	12.0%	6		5.376	-10.4%
w_4	7.812	-2.4%	8		7.89	- 1.3%
w_5	8.778	-12.2%	10		10.626	6.3%
w_6	9.702	-19.2%	12		13.566	13.1%

In M_9, the 1 – 9 scale exceeds the actual maximum weight ratio $w_6/w_1 = 6$. As a result the eigenvector scale overestimates the heavier weights and underestimates the lighter weights. In M_3, the 1 – 3 scale falls short of the maximum weight ratio $w_6/w_1 = 6$. As a result the eigenvector scale underestimates the heavier weights and overestimates the lighter weights. The spread between the two extremes is too large with a scale of 1 – 9 and too small with a scale of 1 – 3. However, all three scales provide the proper ordinal ranking of the weights.

As a practical test of our theory we duplicated Saaty's weight test on five dissimilar objects of various sizes, shapes, and weights:

1. Ski Boots	8 lb. 3 oz.
2. Radio	2 lb.13 oz.
3. Iron	3 lb. 4 oz.
4. Jug of Wax	7 Lb. 9 oz.
5. Pile of Kindling	6 lb. 4 oz.

Pairwise comparisons of these objects were made using the linguistic 1 to 9 scale and the corresponding reciprocal matrix was generated. The results as compiled below are distorted significantly from the actual weight distribution. Nevertheless the eigenvalue $\lambda = 5.30$ is rather low. Again, this low eigenvalue indicates consistency of the responses – not necessarily accuracy of the predictions.

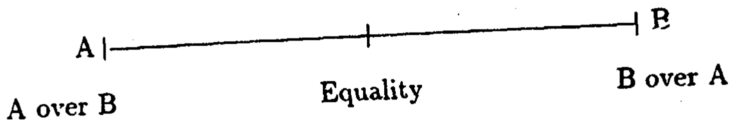
On the other hand if we observe that the maximum ratio is Boots/Radio = $8.1875/2.8125 = 2.9111 \simeq 3$, we see that a maximum ratio of 3 (as opposed to 9) might have been better. Rescaling the original observations linearly to 1 – 3 from 1 – 9 changes the results considerably. These results too are tabulated below; they are seen to be much more acceptable.

WEIGHT TEST

	Scale 1 – 9			Scale 1 – 3	
	Computed Weight	% error	Actual Weight	Computed Weight	% error
w_1	8.7	6.4%	8.1875	7.58	– 7.3%
w_2	0.84	– 70.1%	2.8125	2.75	– 2.1%
w_3	1.4	– 56.9%	3.25	3.06	– 5.8%
w_4	11.79	56.0%	7.56	8.47	12.0%
w_5	5.33	– 14.7%	6.25	6.23	– 0.3%

Using the correct scale reduces the maximum relative error from 70% to 12%; the error at the volatile low end was reduced from 70% to 2%. Note that the original 1 – 9 responses were just rescaled to a 1 – 3 scale. The experiment was not redone. Had the original experiment been redone with a 1 – 3 limitation on responses, much of the "fine tuning" of the responses would have been lost; i.e., not enough linguistic variation would have been permitted.

An alternative to 1 – 9 responses (or any numerical response for that matter) is to use a bar graph in which the center position represents equality of the samples and the ends represent extreme dominance of one sample over the other:



Using such a bar graph, responses can be interpreted on any numerical scale desired. We used bar graphs of this type in an experiment designed to test the ability of panelists to ascertain small differences in samples when the total magnitude was also small. In short, we wanted to test the applicability of this process to situations in which minor differences must be determined; can the process be "fine tuned" to indicate detailed differences as well as general relationships? Again, we tested the process in a situation where the property in question could also be objectively measured. Without such tests, the process would have little credibility in purely subjective situations. The experiment and its results are described below.

<u>An Experiment:</u> The thickness of paper is called "caliper" in the paper industry. Caliper is usually determined under laboratory conditions using instruments capable of accuracy to within 10^{-5} inches. Because of the non-uniformity of any given piece of paper, the caliper is usually measured in several spots and an average of these is used as the caliper of that sample. Thus caliper is an imprecise (even fuzzy) measurement made on a given sample of paper which is representative of the run of paper from which the sample was obtained. Since fuzzy sets provide a framework in which one can study subjective judgments, we attempted to determine how closely the determination of caliper of paper, made by subjective decisions of panelists, compares with the instrumental measurements of the same samples under laboratory conditions.

In all, 29 panelists participated in evaluating 6 paper samples. The papers were chosen so that caliper was essentially the only difference between them. Also to prevent other factors from influencing the results the samples were glued down to uniform metal blocks. The paper was trimmed to within 1/4" of the edge of the block on all sides. Thus "flexibility" for example could not affect the evaluation.

Interpreting the panel responses on the bar graphs in the traditional 1 – 9 linguistic scale gives the following results. The results presented are averages over 27 panelists (2 were eliminated from consideration because of high eigenvalues). Caliper is given in 1/1000's of an inch.

CALIPER TEST
Scale 1 – 9

Paper Sample	Average Caliper	Computed Caliper	Relative Error
1.	3.26	1.80	–44.7%
2.	6.35	4.99	–21.4%
3.	5.63	4.03	–23.1%
4.	10.24	13.04	27.4%
5.	4.29	3.99	7.0%
6.	8.03	9.94	23.8%

Note that almost all errors are large but that the largest error occurs at the low end and that the next largest error occurs at the high end. The sizes of these errors would seem to severely limit the applicability of the process to situations in which such delicate differences occur and are to be detected. On the other hand, the maximum ratio in average measured calipers is 10.24/3.26 $\simeq$ 3.14. Reinterpreting the original panel data on a 1 – 3.14 scale improves the results significantly. The results are tabulated below. An extra column is included to indicate the variations in the several caliper measurements taken to obtain the "Average Caliper" for the given paper sample.

CALIPER TEST
Scale 1 – 3.14

Paper Sample	Average Caliper	Computed Caliper	Relative Error	Inherent Variation in Sample
1.	3.26	3.09	–5.2%	± 4.3%
2.	6.35	5.99	–5.7%	± 8.7%
3.	5.63	5.17	–8.2%	± 6.2%
4.	10.24	10.51	2.6%	± 1.3%
5.	4.29	4.48	4.4%	± 4.7%
6.	8.03	8.55	6.5%	± 3.7%

By rescaling the interpretation from 1 – 9 to a 1 – 3.14 scale, the distortion at the low and high ends has been removed. In samples 2 and 5, the error is no larger than the variation inherent in the sample itself; there is thus effectively no error in our computed caliper. In no case is the error more than double the variation in the sample. We think this kind of accuracy obtained from subjective non-quantified judgments is astounding. It should be pointed out that sample variation does seem to be related to the resulting errors in computed caliper. The lowest sample variation (1.3%) corresponds to the lowest experimental error (2.6%). Thus it would seem that half the experimental error is attributable to variation in the samples themselves. The error generated by the subjective evaluations and the subsequent computational process is no greater than that which is contributed by variations within the samples.

We remind the reader again that we are not proposing that this process be used to measure objectively quantifiable properties. Rather, we are testing our theories on objectively quantifiable properties, such as weight and caliper, so as to lend credibility to the process when it is used in situations which are primarily subjective.

Another Experiment. The determination of print quality has been a subject matter for which many sophisticated instrumental approaches have been developed, but human perception is still used as an integral part of the final evaluation. In [2] non-impact printer image qualities were studied using paired comparisons elicited from panels. Linguistic expressions and graphic responses were both used for transcribing the panel responses. The responses were analyzed using the techniques described in this paper. The purpose of the experiment was to test the applicability of the process outlined in this paper to situations in which minor differences must be determined. The resulting eigenvalue indicated that the panelists were able to give consistent responses in making the paired comparisons; panel fatigue and/or confusion was not a problem. The eigenvector analysis of the panel responses, when averaged, gave results consistent with what one would expect from viewing the samples. Samples which were ranked as being near in quality required higher levels of magnification before significant differences were observed than did samples which were ranked as being far apart in quality. While instrumental measurements made on greatly magnified images can result in overly stringent purchasing requirements, panel testing brings the determination of print quality closer to the practical marketing situation.

Summary and Conclusions. Many applications in commerce and industry demand far greater relative accuracy than is sometimes evidenced in the use of the paired comparison technique to elicit fuzzy properties. Early indications obtained in a fuzzy analysis should in many cases be refined. For instance, the difference between a 1% and a 3% market share could mean a tripling of business volume for a small contender. More precise results would be requested.

Many arguments can be given to justify the utility of a 1 – 9 linguistic scale for subjective responses. However, we have found that it is not necessarily advisable to continue to use the 1 – 9 scale in the computational process which follows the subjective evaluations. In fact, the choice of computational scale greatly influences the results at the extremes - the low and high ends - with related distortions in between. In short, the choice of computational scale dictates (approximately) the ratio between the extremes of the measured property in the given samples as generated by the eigenvector. Too large a scale results in the extremes being separated too far; too small a scale brings the extremes too close together.

Taking greater care in making the comparisons cannot correct for this distortion if an inappropriate scale is chosen. This distortion is inherent in the computational process.

This observation has a theoretical basis as well. For each column of the matrix (when normalized) is an approximation to the (normalized) dominant eigenvector. Thus the maximum ratio in any column will approximate the maximum ratio in the eigenvector. In the column corresponding to the lightest (thinnest, etc.) sample all entries will be integers on a linguistic scale or at least greater than or equal to 1 on a continuous (bar graph) scale. Thus the maximum ratio in this column should be approximated by the ratio of the extremes in the resulting eigenvector. This observation is also borne out in the examples and experiments described in the body of this paper.

Thus in any panel test involving paired comparisons there are two distinct problems :

1. Using an acceptable scale for the panel responses. Whether this scale be linguistic, continuous or otherwise, it would seem that 9 levels on a scale of $1-9$ is perfectly acceptable. An exception to this rule is that if the samples are too diverse, a hierarchical analysis would be in order.

2. The choice of computational scale should be treated as being independent of the scale used by the panelists. If the results are to be realistic and if the accuracy is to be "fine tuned", the computational scale must be close to the actual ratio between the properties in the extreme samples. This is not a trivial problem, however. If a fuzzy analysis is indeed necessary, then presumably this ratio cannot he obtained objectively.

Nevertheless, since some kind of yardstick is probably desired, industry or marketing experts could indicate that one or several scales may be "appropriate". An advantage of the bar graph approach is that it readily lends itself to arbitrary computational scales. Thus results could quickly be processed for several scales and comparisons made. Then other scales could be checked - all without requiring further input from panels.

However, some outside judgments will be necessary in selecting a scale. For the computed spread between the extreme samples is an increasing function of the scale. There is no critical point or critical value which says that in trying several scales, we passed through the correct scale. The only critical point occurs with the minimum scale of 1 which pushes all data together and makes no distinctions whatsoever.

* Department of Mathematics, University of Dayton, Dayton, Ohio
** Boise Cascade R&D, Portland, Oregon
(Now at the Fine Paper Division of Union Camp Corporation, Franklin, Virginia)

BIBLIOGRAPHY

[1] R.E. Bellman, Introduction to Matrix Analysis, second edition, McGraw – Hill, New York, 1970.

[2] P. Lim, R.C. Steinlage, and T.E. Gantner, Application of fuzzy set theory for evaluating non-impact printing image qualities, Proceedings of the 1990 International Printing and Graphic Arts Conference, Vancouver BC, Canada, November 1990, pp. 79 - 83.

[3] G.A. Miller, The magical number 7 ± 2: some limits on our capacity for processing information, Psychological Review, vol. 63, 1956, pp. 81 - 97.

[4] T.L. Saaty, A scaling method for priorities in hierarchical structures, Journal of Mathematical Psychology, vol. 15, 1977, pp. 234-281.

[5] T.L. Saaty, Exploring the interface between hierarchies, multiple objectives, and fuzzy sets, Fuzzy Sets and Systems, vol. 1, no. 1, 1978, pp. 57 - 68.

[6] T.L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw – Hill, New York, London, 1980.

[7] T.L. Saaty, Decision Making For Leaders: The Analytical Hierarchy Process For Decisions in A Complex World, Lifetime Learning Publications, Belmont, CA, 1982.

[8] T.L. Saaty and L.G. Vargas, The Logic of Priorities: Applications in Business, Energy, Health, and Transportation, Kluwer – Nijhoff, Boston, 1982.

NON-SCALAR UNCERTAINTY
Uncertainty in Dynamic Systems

Salvador Gutiérrez Martínez
Instituto Tecnológico de Morelia
Ave. Tecnológico 1500, Lomas de Santiaguito
58120 Morelia, Michoacán.
México.
FAX (451) 216-43

Abstract

The following point is stated throughout the paper: Dynamic systems are usually subject to uncertainty, be it the unavoidable quantic uncertainty when working with sufficiently small scales or when working in large scales uncertainty can be allowed by the researcher in order to simplify the problem, or it can be introduced by non-linear interactions. Even though non-quantic uncertainty can generally be dealt with by using the ordinary probability formalisms, it can also be studied with the proposed non-scalar formalism. Thus, non-scalar uncertainty is a more general theoretical framework giving more insight about the nature of uncertainty and providing a practical tool in those cases in which scalar uncertainty is not enough, such as when studying highly non-linear dynamic systems. This paper's specific contribution is the general concept of non-scalar uncertainty and a first proposal for a methodology. Applications should be based upon this methodology. The advantage of this approach is to provide simpler mathematical models for prediction of the system states.

Present conventional tools for dealing with uncertainty prove insufficient for an effective description of some dynamic systems. The main limitations are overcome abandoning ordinary scalar algebra in the real interval [0,1] in favor of a tensor field with a much richer structure and generality. This approach gives insight into the interpretation of the Quantum Mechanics and will have its most profund consequences in the fields of elementary particle physics and nonlinear dynamic systems. Concepts like "interferring alternatives" and "discrete states" have an elegant explanation in this framework in terms of properties of dynamic systems such as strange attractors and chaos.

The tensor formalism proves specially useful to describe the mechanics of representing dynamic systems with models that are closer to reality and have relatively much simpler solutions. It was found to be wiser to get an approximate solution to an accurate model than to get a precise solution to a model constrained by simplifying assumptions. Precision has a very heavy cost in present physical models, but this formalism allows the trade between uncertainty and simplicity.

It was found that modeling reality sometimes requires that state transition probabilities should be manipulated as nonscalar quantities, finding at the end that there is always a transformation to get back to scalar probability.

Introduction

About 60 years ago, after strong experimental evidence in the field of elementary particle physics, it was realized that probability theory as defined by that time was insufficient to handle the unavoidable uncertainty in the behavior of microscopic physical systems. As stated by the late R.P. Feynmann "...the laws of probability which are conventionally applied are quite satisfactory in analyzir ne behavior of a roulette wheel but not the behavior of a single electron or a photon of light." - *"Quantum Mechanics and Path Integrals"*, R.P.Feynmann & A.R. Hibbs, McGraw-Hill, 1965. As a result various formulations of theories generally known as Quantum Electrodynamics and Quantum Mechanics were born. These theories have proven to be enormously successful as predictive tools and are in this respect unchallenged to this day, though they have originated much controversy by their philosophical implications. Nevertheless, all of them overcome the limitations of probability to deal with the results of experiments. They do so because they invariably recur to algebraic structures much richer than the real interval [0,1]; all of them involve first working with complex or hypercomplex fields and multidimensional structures and then prescribe a transformation that restates the predictions in conventional probabilistic terms. It is not in the scope of this paper to state a formulation or description of any of these theories, for there are countless of them available in the subject's literature. They are mentioned as a monumental example of the potential of multidimensional structures and complex fields in the treatment of uncertainty.

The higher dimensionality and more complex operations involved in complex and hypercomplex fields are useful to generate the predicted patterns in the probability distributions of interferring alternatives.

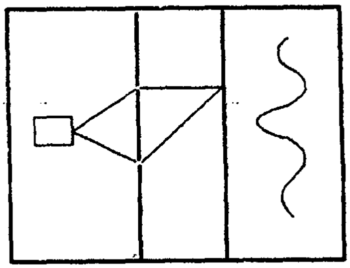


Figure 1 *Young's Experiment*

What are interferring alternatives can be illustrated by Young's experiment (Figure 1), which is in this description a thought experiment that can be instrumented in more realistic settings. A source of particles (electrons or photons, whatever), emits them toward a screen, but between the source and the screen we place a barrier with two slits. If we make the beam so weak that it consists of a single photon at a time, we could assume that a single particle would go through either slit and then it would be recorded at the screen. After a great number of particles have made their way one by one through the screen they would form a visible pattern on the surface which would represent the relative frequency (probability) distribution of a particle coming from the source, through the slits, reaching a certain point on the screen. This probability density function is represented by the curve at the right of the screen in Figure 1 and is unexpected, since it has many local maxima and minima as if it were recording the effects of waves instead of particles.

If we could know with certainty that electrons come through either slit say, by blocking one of them, then we would record a probability density function more like the curve indicated in Figure 2. A similar result would be found when the other slit is blocked (Figure 3). If both distributions were independent from each other we would find that the probability density function that would be the sum of the previous two ones, giving a bell shaped curve.

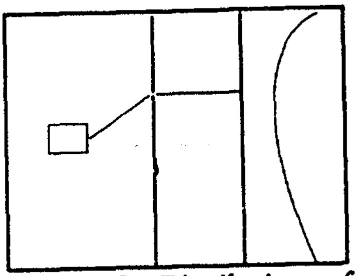


Figure 2 *Distribution of Particles through one Slit.*

Nevertheless, the observed frequency distribution is very different from the expected one (Figure 4). This is what is meant by saying that the alternatives ***interfere***, much in the way waves do, but the interference pattern is defined on the probability (frequency) distribution. Whenever two or more alternatives cannot be resolved by experiment, they always interfere. The difference between the observed and expected patterns is caused by the fundamental uncertainty described by Heisenberg's principle. Whenever we want to interact with the particles to find out which way they came the interference pattern at the screen is blurred. If we would like to determine the particle's path by getting it to interact with a photon or some other particle, then the disturbance produced by the sensing particles would be unavoidably too big to find out with precision where was that particle going, thus destroying the interference pattern.

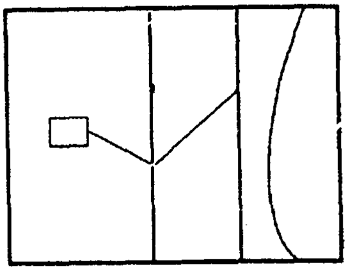


Figure 3 *Distribution when the other Slit is Closed.*

Dynamic Systems Subject to Uncertainty

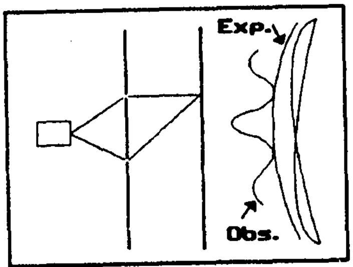


Figure 4 *Observed vs. Expected Frequency Dist.*

The essence of dynamic systems is time dependency. When observing microscopic dynamic systems we can say that much of Heisenberg's unavoidable uncertainty can be focused on the time variable, since most of the relevant variables are time dependent. In this kind of dynamic systems it is not possible to say with arbitrary precision that a given particle is in a certain well defined state at any precise moment, nor is it possible to say that it has a defined trajectory and there exist sets of time dependent variables whose precise value can not be known simultaneously, such as position and momentum. A similar argument holds for <u>any kind</u> of dynamic system subject to uncertainty, specially for non-linear ones. This means that we are left with a system which can assume a set of states which can be either well defined or fuzzy, but we do not know in what state will the system be at a certain time.

Because of this loss of information on time dependency we are forced to study dynamic systems disregarding the time variable; i.e., we are compelled to make <u>time independent statements about the states of a system</u>. This is just the kind of statement that a frequency (probability) distribution is: What states can a dynamic system assume and how likely is it to be in any of them given some determined boundary conditions. Because of uncertainty, observing the system by means of an experimental setting means that we will not generally observe the same outcome for identical repetitions of the experiment. So, all we can do is repeat experiments and measurements for a large number of times and then watch the relative frequency of the outcomes. Theoretical statements can only be made in probabilistic terms and confrontation with reality can only be made in terms of comparing predicted probabilities against observed relative frequencies.

The theory of dynamic systems shows that there are some states that can be called "equilibria", which means that once a system has reached one of them it tends to stay in it for a long time. If a system tends to abandon an equilibrium state at the slightest perturbation then this point is called "unstable". Of course, if we observe just the opposite, i.e. the system tends to stay in some state regardless the effect of small perturbations, then it is called a "stable equilibrium". Of course, things in reality are not always that simple for we can find some special states around which the behavior of the system tends to wander. They are called *strange attractors* and can have a simple or very complex nature. The reader is referred to the vast literature on the subject to extend and clarify these concepts.

If a dynamic system subject to uncertainty has strange attractors, they will show up in the frequency (probability) chart as a peak, band or concentration of points, since it will spend a considerable part of the time on them. These peaks look very much like interference patterns when the dynamic system is defined by non-linear functions.

This point can be nicely illustrated with a very well known example, the Verhulst Process (a population growth model, [Peitgen]). We make the following initial assumptions:

$$x_0 \equiv \textit{Initial Population Size}$$

$$x_n \equiv \textit{Population Size after n years}$$

$$R = \frac{(x_{n+1}-x_n)}{x_n} \equiv \textit{Relative Increase per year}$$

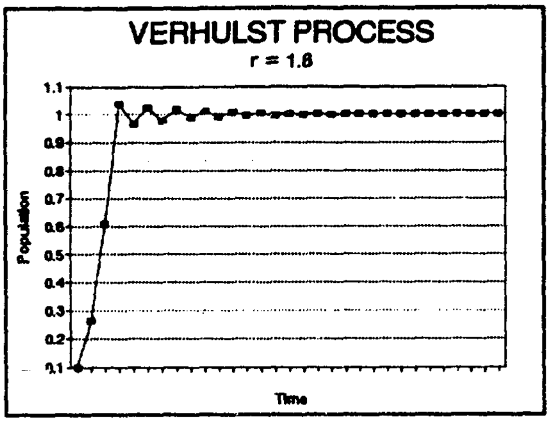
If this rate is constant (say 'r'), then the law is:

$$x_{n+1} = f(x_n) = (1+r)x_n$$

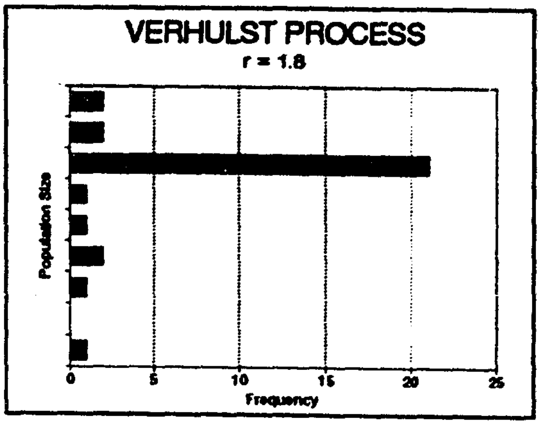
If **R** varies with population size, then **R** = $r(1-x_n)$, where $r > 0$ is the "growth parameter". Then,

$$x_{n+1} = f(x_n) = (1+r)x_n - rx_n^2$$

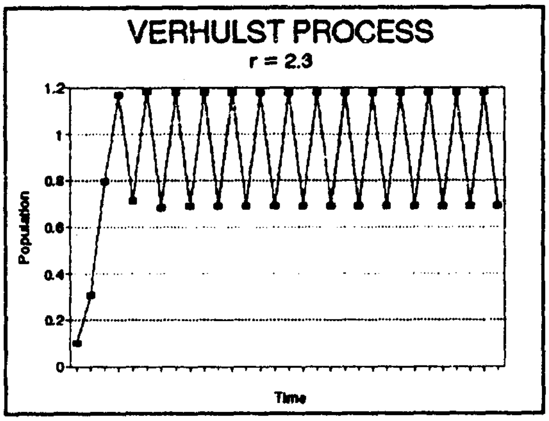
Then $x_0 = 0$ and $x_0 = 1$ are equilibrium points. Analysis for $0 < x_0 << 1$, $r > 0$ yields:



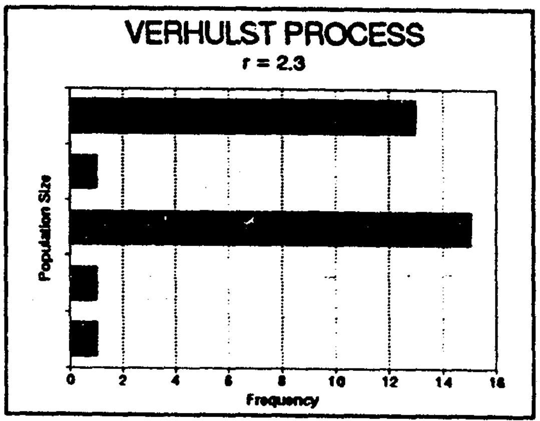
Ilustr. 5 *One Attractor*



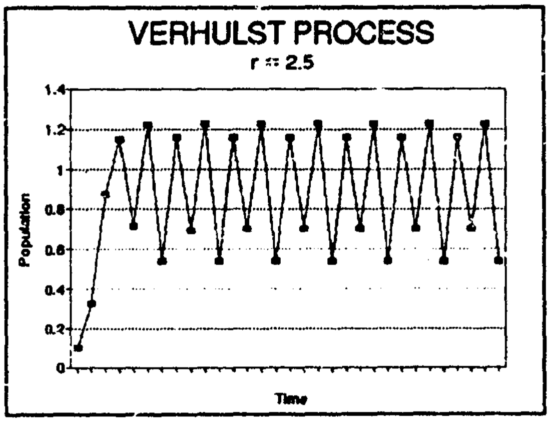
Ilustr. 6 *Interference Pattern*



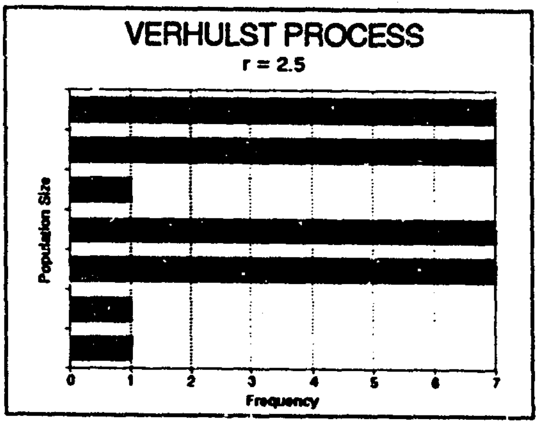
Ilustr. 7 *Two Attractors*



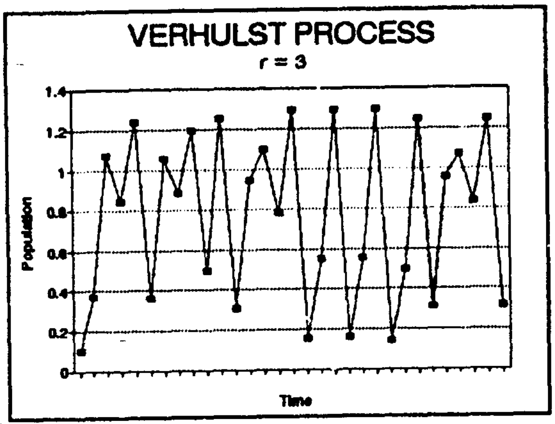
Ilustr. 8 *Interference Pattern*



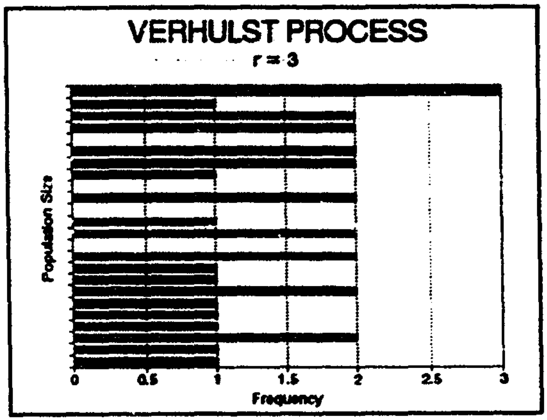
Ilustr. 9 *Four Attractors*



Ilustr. 10 *Interference Pattern*



Ilustr. 11 *Chaotic Behavior*



Ilustr. 12 *Interference Patterns in the Frequency Distribution*

As the Frequency Maps show (Figures 6,8,10 & 12), there are "interference" patterns apparent in the frequency distribution of states. As the growth parameter r increases the behavior becomes chaotic, but the "interference" still shows up.

State Space

State Space can be regarded as the "arena" where dynamic sytems perform and leave their trails and is defined as the set of all possible states a certain dynamic system can attain. The definition of state space presumes the definition of state variables and supports and their value sets. Reaching this stage is equivalent to climbing the first rung in a ladder of epistemologic levels [Klir 1, pp. 16, 33-64], defining a Source System; i.e., isolation of a system from reality to the point where everything is ready to perform observations and to get data.

Definition of state variables and their nature almost defines the nature of state space. It only leaves now to define some other general properties of such a space -i.e., metrics, continuity, compacity, discreteness, order, invariance requirements, etc.

There is no reason to suppose that any two dynamic systems should have the same state space, not even two distinct Source System definitions from the same dynamic system. This is why we need to define the essential properties which a state space should have in order to reach a meaningful methodology.

1.- State space is a metric space $S=(X, \delta)$, where X is a set of elements (points) and δ is a distance function satisfying

a) $\delta(x,y)=0 <=> x=y, \ x, y \in X$,

b) $\delta(x,y) = \delta(y,x), \ x, y \in X$,

c) $\delta(x,z) \leq \delta(x,y) + \delta(y,z), \ x,y,z \in X$

Cover Space

To ensure an adequate structure we regard the state space S as a subspace of another one called C (Cover Space), which is linear, of the same dimensionality as S and has a scalar product that agrees with the metric defined on S.

Quantic Uncertainty

Quantum Theory formulations can be made equivalent to some extent. Basically, we can say that states of a system correspond to vectors in a specially defined Hilbert space [Dirac, p. 51] or to wave functions [Landau, p. 19]. Linear operators defined on such spaces can be made to correspond to dynamic variables. If such operators satisfy some other requirements -being Hermitian or self-conjugate- they can be made to correspond to physical "observables", or quantities that can be measured or observed. Furthermore, if they have eigenvalues and eigenvectors, then they are real quantities and can be interpreted as the values assumed by the dynamic variables associated to such operators when the system is in the state corresponding to those vectors. States can now be "superposed" and probability distributions for states and values can be obtained by first finding the square of the modulus of the associated vector or wave function amplitude and then getting its square root.

Quantum phenomena are subject to an unavoidable and intrinsic kind of **uncertainty** manifest at atomic scales, stated first by W. Heisenberg in the 1920's. Such uncertainty is responsible for the unexpected results in experiments such as Young's -described previously in this paper. It enters the formalisms where we would expect to find it; i.e., associated in some way to the time variable, which is "translated" to phase components of certain complex numbers. In the limit case when Quantum Mechanics approaches Classical Mechanics wave functions can be seen to have terms of the form

$$ae^{\frac{i}{\hbar}S}$$

In this limit case, phase is proportional to S, the mechanical action of a system. But *h* (Planck's constant) implies meaningful contributions (ones which do not cancel out) only at the microworld level scales. Thus, quantum mechanics formalisms introduce the effects of this unavoidable uncertainty into the phase component of a complex quantity. This allows for addition of these complex quantities over "all possible paths" in transition from one state to another, where paths which are very close to each other contribute constructively to the final amplitude, whereas unlikely paths which require relatively much longer times, tending to cancel their contributions out. This is R. Feynmann's approach [Feynmann 1, pp. 31-38]. It should be stressed that these processes operate on **non-scalar** elements all the time. It is only at the end when one can translate that complex amplitude to a probability via the scalar product operation, when finding the modulus of the resulting amplitude vector. It is very important to realize that quantum dynamic phenomena exhibiting interference patterns in frequency or probability distributions **cannot** be explained when manipulating uncertainty as a scalar quantity. The non-scalar nature of amplitudes is what allows constructive and destructive contributions when adding them up. Thus, we may confidently state that uncertainty can be represented and manipulated as a non-scalar quantity in a manner which is perfectly consistent with the conventional scalar probability framework. This at least, is the case in one of the most successful scientific paradigms

Non-scalar Uncertainty

It should be realized that at least in Quantum Mechanical phenomena we find interference patterns in probability or frequency distributions and these patterns are blurred out when introducing the unavoidable uncertainty associated to observations at these scales. In the same fashion, we find this kind of patterns in probability or frequency distributions in the macroscopic world, specially when studying non-linear dynamic systems -c.f. the Verhulst Process described before. And they can be blurred, too, but this time by observer-introduced uncertainty in the measuring processes or by error propagation in calculations.

It is not difficult to see that the additive and monotonic properties of all existing scalar uncertainty frameworks (probability, possibility, etc.), make it impossible to model the destructive contributions from events which are independent from the point of view regarding their origin, but which can interact in strong ways (as is the case in non-linear phenomena), resulting in an overall decrease in the probability of the union of **both** events. In other words, interference patterns in probability distributions cannot be constructed from scalar probability quantities by ordinary means.

One of the main claims of this paper is that there are many instances in ordinary practical situations (especially when dealing with non-linear dynamic systems), where the following requirements are due:

I) If A_i are arbitrary events

$$P(A_1 \cup A_2 \cdots \cup A_n) \le P(A_1)+\cdots+P(A_n) - \sum_{i_1 \neq i_2} P(A_{i_1} \cap A_{i_2}) + \sum_{i_1 \neq i_2 \neq i_3} P(A_{i_1} \cap A_{i_2} \cap A_{i_3}) - \cdots$$
$$+ (-1)^{k+1} \sum_{i_1 \neq i_2 \neq i_k} P(A_{i_1} \cap A_{i_2} \cap \cdots A_{i_k})$$

II) This probability should also "fluctuate" along some support variables

$$P(A_1 \cup A_2 \cup \cdots \cup A_n) = \varphi(t_1, t_2, \cdots, t_m) \quad \textit{where } \varphi \textit{ is}$$

an arbitrary function; the t's are system variables

In practical situations this can be an empirical function sampled in some point/intervals of the domain. Instances of it are interference patterns in the screen in Young's experiment, or the bar graphs along the line graphs illustrating the Verhulst process previously described.

In other words, $\varphi(t_1, t_2, \cdots, t_m)$ is a marginal probability density function which could be obtained by the projection of an (m+1)-dimensional joint probability density function $\phi(t_1,\cdots,t_p,\tau_1,\cdots,\tau_m)$. This process can be called "collapsing" the support variables. This "collapse" can be accomplished in many ways. By introducing the time variable and the effects of uncertainty as phase components (temporarily adding one dimension) and then obtaining the (squared) modulus by a scalar product (eliminating the time variable), as in quantum mechanics formalisms. Or by first finding the joint probability density function empirically, introducing the effects of uncertainty along with the time variable and then projecting over the state variables (collapsing the support variables). This can be stated also as a contracted product if ϕ is regarded as something like a tensor.

There are more reasons other than notation that make it convenient to consider these as tensor quantities. Invariance with respect to base changes and basic operational needs make it desirable to define them as tensor-like arrays. Uncertainty can then be viewed as a tensor-like quantity of order zero (scalar), order 1, 2,..., etc.

If we define the state of a system under the very general frame described by G.J. Klir [Klir 1], a general method to manage the effects of uncertainty can be described as first concentrating these effects on the support variables by "coarsening" their resolution (reducing the number of possible states of these variables and/or broadening the sampling intervals), allowing a better determination of the true system variables and then collapsing all support variables, leaving only support independent frequency or probability distribution functions. This is only one way to take advantage of the uncertainty/simplicity trade pointed out by Klir [Klir 2]. A first contribution is that by regarding the system states and the frequency and probability distributions as tensor-like quantities we get invariance to changes of base and some operational advantages inherited from their new algebraic status. Thus, a state of the system, a vector whose components are the values of the system variables suitably defined by a methodology like Klir's [Klir 1], becomes S_i, a subspace of C_m, the cover space, $1 \leq i \leq m$, where m is the number of state variables excluding supports. The overall system behavior array, with as many dimensions as system variables and supports, and ones in those elements which correspond to observed overall states is

$$B_{\underset{i}{\times} S_i}^{\tau_1,\cdots,\tau_n} \textit{ where } \tau_k \textit{ is the } k^{th} \textit{ support}$$

Here, $\underset{i}{\times} S_i$ is the cartesian product of all the state variables' value sets. If $\mathbf{1}_{\tau_1,\cdots,\tau_n}$ is an all ones array, then the unnormalized frequency distribution function becomes

$$\acute{\phi}_{\underset{i}{\times S_i}} = B^{\tau_1,\ldots,\tau_n}_{\underset{i}{\times S_i}} \cdot 1_{\tau_1,\ldots,\tau_n}$$

this is the joint frequency distribution of all states. The dot indicates generalized matrix product reducing over the repeated indexes. Of course, we can leave some supports in and some out, obtaining the corresponding joint frequency distribution:

$$\acute{\phi}^{\tau_{n_k}\ldots\tau_{n_r}}_{\underset{i}{\times S_i}} = B^{\tau_1,\ldots,\tau_n}_{\underset{i}{\times S_i}} \cdot 1_{\bar{\tau}_{n_k},\ldots,\bar{\tau}_{n_r}} = \acute{\phi}^{\tau_1\ldots\tau_n}_{\underset{i}{\times S_i}} \cdot 1_{\bar{\tau}_{n_k}\ldots\bar{\tau}_{n_r}}$$

where $\bar{\tau}_{n_k}\cdots\bar{\tau}_{n_r}$ means "include all indexes except the barred ones".

To normalize the frequency distributions we divide each element by the scalar

$$\nu = \acute{\phi}^{\tau_1\ldots\tau_n}_{\underset{i}{\times S_i}} \cdot 1^{\underset{i}{\times S_i}}_{\tau_1\ldots\tau_n}$$

which is the total number of observed states, so

$$\phi_{\underset{i}{\times S_i}} = \frac{1}{\nu}\acute{\phi}_{\underset{i}{\times S_i}}$$

and we have

$$\phi_{\underset{i}{\times S_i}} \cdot 1^{\underset{i}{\times S_i}} = 1$$

State Transition Uncertainties

Transition from one state to another involves computing T(x,y), a function expressing the *difficulty* of going from state x to state y in terms of the time variable. I.e., it must be proportional to the *time* taken to go from one state to the other. Then, we can define a two-dimensional quantity

$$\pi(a,b) = \sum_{a}^{b} a e^{iT(x_i,y_i)}$$

which represents the non-scalar likelihood to go from state a to state b. If time is not the only support we wish to "collapse" then we need another component in this vectorial (complex or hypercomplex) quantity. All quantities can be normalized so that we can get real probabilities when obtaining their norms by means of scalar products. Finding function T means we know the state transition structure of the system and we can relate it to time or other support variables. An important comment about the state-transition likelihood is that the sum is computed along *all possible paths* from a to b in such a way that those paths which differ very little from each other have a more important contribution to the final non-scalar transition likelihood. So we can say that there are *preferred* paths in any system.

It is convenient to express all these quantities with complex or hypercomplex numbers, but it is clear that they can be represented in other algebraic settings.

In the simple example of Young's experiment, we can simplify T to be proportional to the length of the paths followed by the particles, then it is evident that we should get an interference pattern, since uncertainty can be referred to a distance (wavelength), too.

Conclusion

This paper contributed the concept of general non-linear uncertainty and a proposed methodology to deal with it. The advantages of using it are a simplification of mathematical models due to the controlled admission of uncertainty.

Dynamic Systems subject to uncertainty are cases where ordinary treatment and calculus of uncertainty is not enough to provide an adequate description of the system. Therefore, a more general and powerful calculus is needed where scalar algebra in the real interval [0,1] is replaced by a complex or hypercomplex field, which have a much richer structure and generality. This calculus is homeomorphic to the methods of Quantum Mechanics and its study and development throws much light on foundational issues of Quantum Mechanics and the now available mathematical tools for managing uncertainty. Also, phenomena such as "interferring alternatives" so basic to Quantum Mechanics find a very elegant explanation in this framework.

References

1.- [Feynmann 1] R.P. Feynmann & A.R. Hibbs. *"Quantum Mechanics and Path Integrals"*, McGraw-Hill.

2.- [Landau 1] L. Landau & E. Lifshitz. *"Mecánica Cuántica"*, Mir.

3.- [Dirac 1] P.A.M. Dirac. "*Mecánica Cuántica*", Ariel.

4.- [Klir 1] G.J. Klir. "*Architecture of Systems Problem Solving*", Plenum 1986.

5.- [Klir 2] G. J. Klir and T. Folger. "*Fuzzy Sets and Information Theory*". 1989.

6.- [Peitgen] H.-O. Peitgen and P.H. Richter. "*The Beauty of Fractals*". Springer-Verlag.

N93
29566
UNCLAS

S10-67
161850

N93-29566

p, 9

Comparison Between The Performance Of Two Classes Of Fuzzy Controllers

Janabi T.H.
Director of Technology
Mentalogic Systems Inc.
7500 Woodbine Avenue
Suite 310
Markham, Ontario
L3R 1A8 Canada

Sultan L.H.
Director of Research & Development
Mentalogic Systems Inc.
Associate Professor
Department of Computer Science
Yourk Unicersity
2275 Bayview Avenue
Toronto, Ontario
M4N 3M6 Canada

September 30, 1992

Abstract

This paper presents an application comparison between two classes of fuzzy controllers; the Clearness Transformation Fuzzy Controller (CTFC) and the CRI-based Fuzzy Controller. The comparison is performed by studying the application of the controllers to simulation examples of nonlinear systems. The CTFC is a new approach for the organization of fuzzy controllers based on a cognitive model of parameter driven control, the notion of fuzzy patterns to represent fuzzy knowledge and the Clearness Transformation Rule of Inference (CTRI) for approximate reasoning. The approach facilitates the implementation of the basic modules of the controller: the fuzzifier, defuzzifier and the control protocol in a rule-based architecture. The CTRI scheme for approximate reasoning does not require the formation of fuzzy relation matrices yielding improved performance in comparison with the traditional organization of fuzzy controllers.

1 Fuzzy Logic Controllers

Fuzzy controllers have emerged to the engineering practice as a convenient tool for modelling the operator knowlegde and experience of controlling complex processes and systems. The basic assumption behind their dissimination is the ability to imitate the approximate reasoning mechanisms that the human operator applies to make decisions in complex and vague situations. The great works of L.A. Zadeh [5, 6, 7, 8] on fuzzy reasoning has opened a new avenue for artificial approximate reasoning which is required to intellectualize machine decisions. The compositional rule of inference (CRI) which Zadeh introduced as a tool for approximate reasoning [6] has been successfully applied for the synthesis of linguistic control protocols of skilled operator, thereby making the design of fuzzy logic controllers possible.

However, no systematic approach exists for the design of fuzzy controllers. The main drawback seems to lie in the application of the CRI scheme which requires the formulation of fuzzy relation matrices and the performing of the Max-Min operations associated with them. For complex processes these matrices are multidimensional and the computaion time required to perform the Max-Min operations can go beyond real-time problem solving and control requirements.

Mentalogic Systems Inc. developed a new approach for the design of fuzzy controllers based on the operator cognitive model of fuzzy control [4], and using a new approximate reasoning scheme that requires niether the fuzzy relation matrices nor the Max-Min operations associated with them. This scheme is called the Clearness Transformation Rule of Inference (CTRI). It is a real-time approximate reasoning scheme in which calculations are remarkably reduced in comparison with the CRI.

2 The Operation of Fuzzy Controllers

Fuzzy logic Controllers can be classified as control expert system capable of interpreting fuzzy statements of human knowledge such "pressure is low" or "decrease steam flow slightly" etc. Using the CRI scheme the control actions are deduced by the composition of fuzzy sets generated from the measured values of the process variables (which are the input to the fuzzy controller), and the matrices of fuzzy rules (knowledge on the input-output relationship) using the algebraic operations of the Max and Min. Fuzzy logic controllers map input crisp data into fuzzy linguistic terms described by vectors (fuzzification), deduce the control actions as fuzzy sets in the form of vectors also using the CRI, then translate these actions into crisp data (defuzzification) whic is applied to regulate the controlled process. The overall operation of the fuzzy controllers can be looked upon as numerical mapping procedure in which the compositions of fuzzy sets and fuzzy rules are handled by the CRI while the controller provides numerical to linguistic (fuzzification) and linguistic to numerical (defuzzification) converters to communicate with the controlled process.

The CTFC fuzzy controller, however, is designed following the operator cognitive model of control [1, 2, 3]. It has a modular structure in which each module performs a set of distinct tasks. These tasks are the fuzzification, rule selection, approximate reasoning, and defuzzification. Contrary to the CRI designs of fuzzy controllers which are data processing devices, the CTFC is a cognitive pattern processing device which recognizes fuzzy patterns and processes them to perform its decision making procedure. An overall account of the CTFC controller is as follows. The controller receives crisp data which represents the states of the process variables to be controlled. This data is channeled to the fuzzifier module which recognizes their fuzzy patterns and their clearness assessments in a cognative manner. The output of the fuzzifier is then used by the Domain Knowledge-Base and approximate reasoning module for rule matching and clearness assessment of the fuzzy patterns of the process situation. The defuzzifier then generates the fuzzy control actions which are then translated to control commands in the form of crisp data which is subsequently sent to regulate the process.

In this controller a fuzzy pattern is defined by the triple { S, D, A }, where:

S - is the syntactical description of a fuzzy pattern. The logic of fuzzy predicates is utilized to describe the fuzzy patterns of the real world situations. The notion of a fuzzy predicate as an atomic formula of this logic is considered an elementary fuzzy pattern. Complex fuzzy patterns are described as well formed formulae (WFF) of this logic.

D - is the domain to which the fuzzy pattern is attached. This domain is composed of three attributes:
L_x: is the domain variable.
X: is the space of all instantial models of L_x .
σ_x: is the set of allowable substitutions of the models of X for L_x .

A - is the clearness assessment of a fuzzy pattern. This assessment employs a clearness measure built in the closed interval [0, 1] and divided into a finite number of clearness values (a_k).

Two types of fuzzy patterns are employed by the CTFC controller; The static fuzzy patterns stored in the knowledge-base of the controller, and the dynamic fuzzy patterns denoting the patterns detected in real dynamic operations. The static and dynamic patterns have the same syntactical description but may differ in their clearness evaluation in terms of "strength" and "weakness". Global and local clearness assessments are employed to describe the static and dynamic fuzzy patterns.

2.1 Process Representation

For the simulation examples presented in this paper, the variables which are used to represent the process are the error in the output response and the change of this error. These variables are calculated as in equations (1) and (2) below. The fuzzy predicates utilized for each variable as shown in figure (2). The same fuzzy sets were used for the error, change of error and output. The control rules were different for each control system.

$$e_{present} = 1 - out \tag{1}$$

$$ce = e_{present} - e_{previous} \tag{2}$$

where:
$e_{present}$ = present error in the output response (for a unit step input).
$e_{previous}$ = previous error in the output response (for a unit step input).
ce = change of error in the process response.

The control rules which are used in the fuzzy controller are application dependent. To formulate the control protocol we generally started with some approximate rules, then improved these rules in the direction which improved the controller performance for obtaining better process output response.

For the two variables chosen, the error and the change of error, sixty four rules were sufficient to describe the control requirement for each simulation example.

2.2 Simulation Results

To compare the performance of the CTFC and CRI-based controllers, simulations were performed using the same controlled systems under the same simulation conditions which are achieved by employing the same fuzzy sets and control rules for both controllers. The systems chosen are nonlinear and representing problematic systems from control point of view. Their synthesis reflects the capability and limitation of each controller. The systems are single-input single-output closed loop nonlinear systems with single valued and double valued nonlinearities. Two examples are presented here. The first example involves a single valued nonlinearity, and the second example involves a double valued nonlinearity. Figure (1) shows a block diagram of the closed loop system.

Example 1

In this example, the linear element is a second order system having a free integrator, and described by the transfer function

$$G(s) = \frac{2.5}{s^2 + 0.3s + 0.1}$$

The nonlinear element is on-off plus dead-zone as shown in figure (3). The rules which are used for the control of this system are shown in figure (5). The system response before and after compensation using both the CTFC and CRI-based controllers is shown in figure (4). Both controllers were capable of eliminating the steady-state error caused by the dead-zone. However, the CTFC controller response is much smoother than that of the CRI-based controller. The gain of the CRI-based had to be raised to obtain this response. Lowering the gain to that of the CTFC controller gave zero output response because the controller output always fell within the dead-zone of the nonlinearity. The controller was not capable of emerging outside the dead-zone. The CTFC was capable of addressing this system without requiring any outside interference or help, reflecting better capability and hiher intelligence in handling difficult systems. Note the elimination of steady state error despite the presence of the dead-zone.

Example 2

In this example, the linear element is a second order system having double integrator, and described by the transfer function

$$G(s) = \frac{1}{s^2}$$

The nonlinear element is a backlash nonlinearity as shown in figure (3). The rules which are selected for the fuzzy controller are displayed in figure (7). The system response before and after compensation using both controllers is shown in figure (6). In this system the CTFC controller yielded excellent response while the CRI-based controller failed completely in addressing this system. The superiority of the CTFC over the CRI-based controller is clearly reflected in this example. It is interesting to note that the nonlinear element in this systems is a double valued nonlinearity.

3 Evaluation and Conclusions

A comparison simulation study has been conducted between the CTFC and the CRI-based fuzzy controllers to illustrate the capabilties of each controller in addressing difficult control systems. The systems chosen for the comparison study are nonlinear control systems. One system was chosen with single valued nonlinearity and the other system with double valued nonlinearity. For the comparison to have a meaningful interpretation the same fuzzy sets and control rules were employed in both controllers.

The results of the simulation show a clear advantage of the CTFC controller over the CRI-based controller. The CTFC was capable of addressing both systems giving smooth response for them, while the CRI-based fuzzy controller gave a 25% overshoot in the first system and failed completely in addressing the second system.

The simulation examples also reflect the capabilty of the CTFC fuzzy controller in addressing systems with double valued nonlinear elements, and clearly illustrate the optimum solution embedied in this controller.

References

[1] J., Rasmussen, *"Outlines of a Hybrid Model of the Process Plant Operator"*, In Sheridan T.B., and, Johanssen G., Eds., Monitoring Behaviour and Supervisory Control, N.Y., Planum Press, 1976.

[2] J., Rasmussen, *"A Frame for Cognitive Task Analysis in Systems Design"*, In: Intelligent Design Support in Process Environments (E. Hollnagel et. al. edrs), NATO ASI Series, Vol. 21, pp. 175-195, 1986.

[3] J., Rasmussen, J. *"Information Processing on Human-Machine Interaction, An Approach to Cognitive Engineering"*, Vol. 12, N.Y., North-Holland Series in Science and Engineering, 1986.

[4] L.H. Sultan, and T.H. Janabi," Truth Transformation Fuzzy Logic Controllers: Outlines of the Design of a New Generation of Fuzzy Controllers", AI Conference, Wyoming, 1991.

[5] L.H., Zadeh, "*Fuzzy Sets* ", Information and Control, Vol. 8, pp. 338-353, 1965.

[6] L.H., Zadeh, "*Outline of a New Approach to the Analysis of Complex Systems and Decision Process*", IEEE Trans. on Systems, Man and Sybernetics, Vol. SMC-3,No. 1, pp. 28-44, 1973.

[7] L.H., Zadeh, "*Fuzzy Sets as a Basis for Theory of Possibility* ", Mem. UCB/ERL M77/12, University of California, Berkeley, 1977.

[8] L.H., Zadeh, " *The Concept of Linguistic variable and Its Application to Approximate Reasoning: Parts I and II* ", Inf. Science, 8, pp. 199-249 and 301-357, 1975.

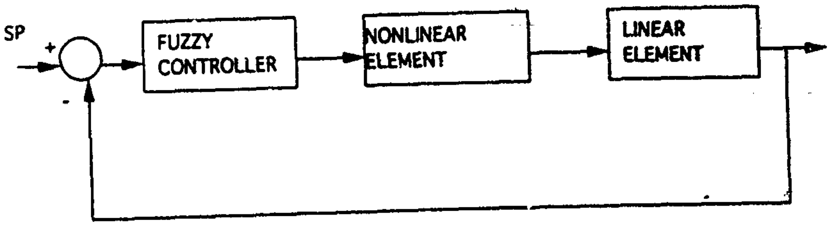


Figure 1. Closed Loop Block Diagram

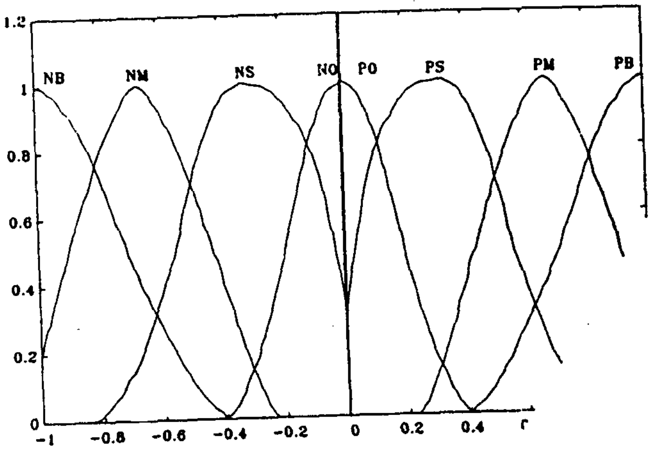
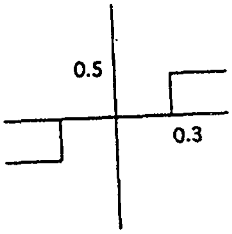
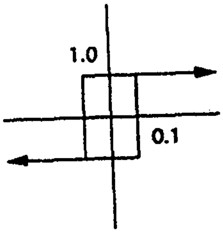


Figure 2. Fuzzy sets



ON-OF plus Dead-zone



Backlash

Figure 3. Nonlinear Elements

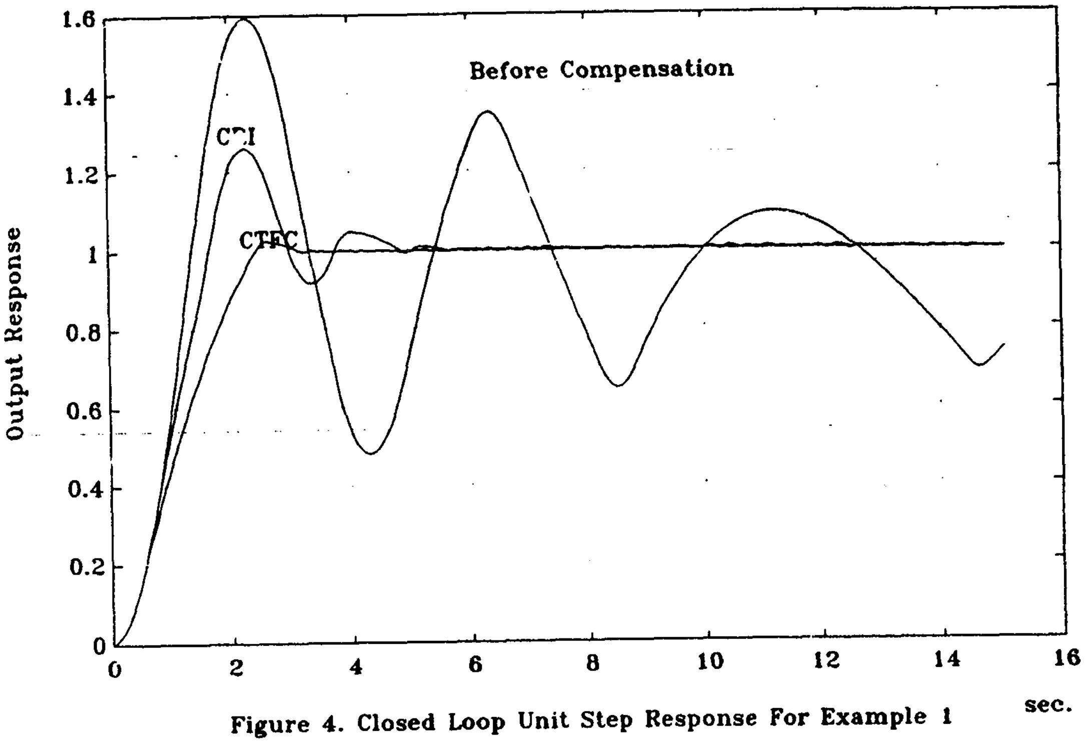


Figure 4. Closed Loop Unit Step Response For Example 1

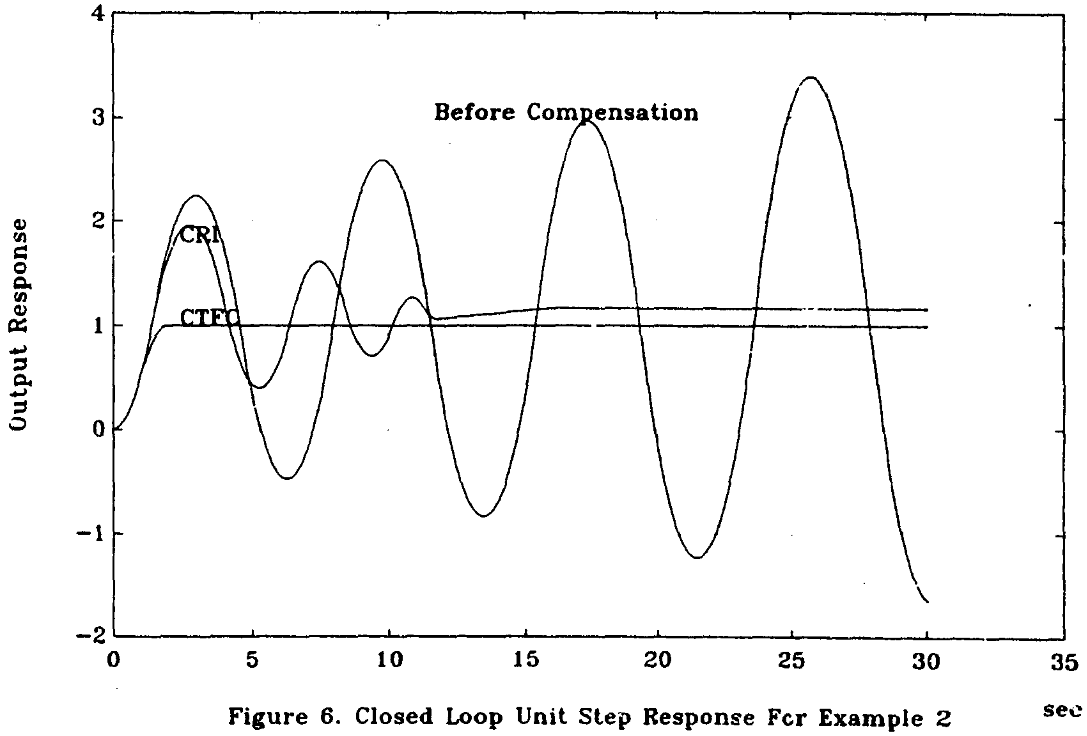


Figure 6. Closed Loop Unit Step Response For Example 2

IF	AND	THEN

E = NB	CE = NB or
	CE = NM
		CA = NB

E = NB	CE = NS or
	CE = NZ or
	CE = PZ or
	CE = PS
		CA = NS

E = NB	CE = PM or
	CE = PB
		CA = NZ

E = NM	CE = NB or
	CE = NM or
	CE = NS
		CA = NB

E = NM	CE = NZ or
	CE = PS
		CA = NS

E = NM	CE = PZ	CA = NZ

E = NM	CE = PM or
	CE = PB
		CA = NM

E = NS	CE = NB c
	CE = NM
		CA = NS

E = NS	CE = NS or
	CE = NZ
		CA = NB

E = NS	CE = PZ or
	CE = PS
		CA = NS

E = NS	CE = PM or
	CE = PB
		CA = NM

E = NZ	CE = NB or
	CE = NM or
	CE = NS
		CA = NS

E = NZ	CE = PZ	CA = NZ
E = NZ	CE = NZ	CA = NB

E = NZ	CE = PS or
	CE = PM or
	CE = PB
		CA = NM

E = PZ	CE = NB	CA = NZ

E = PZ	CE = NM or
	CE = NS
		CA = PZ

E = PZ	CE = NZ	CA = PS

E = PZ	CE = PZ or
	CE = PS or
	CE = PM or
	CE = PB
		CA = PB

E = PS	CE = NB or
	CE = NM or
	CE = NS or
	CE = NZ
		Z

E = PS	CE = PZ o
	CE = PS
	CE =
	CE	

E = PM	C	
E = PM		
E = P		
E =		
E = P		

Figure 5. Control Rules for Example 1

The abreviations used are:
E = Error
CE = Change in Error
CA = Control Action
NB = Negative Big
NM = Negative Medium
NS = Negative
NZ = Negative Zero
PB = Positive Big
PM = Positive Medium
PS = Positive Small
PZ = Positive Zero

IF	AND	THEN

E = NB	CE = NB or
	CE = NM or
	CE = NS
		CA = NM

E = NB	CE = NZ	CA = NB

E = NB	CE = PZ or
	CE = PS or
	CE = PM or
	CE = PB
		CA = NZ

E = NM	CE = NB or
	CE = NM or
	CE = NS or
	CE = NZ
		CA = NB

E = NM	CE = PS or
	CE = PZ or
	CE = PM or
	CE = PB
		CA = NZ

E = NS	CE = NB or
	CE = NM or
	CE = NS
		CA = NB

E = NS	CE = NZ or
	CE = PZ or
	CE = PS
		CA = NS

E = NS	CE = PM or
	CE = PB
		CA = NM

E = NZ
E = NZ
	CE = NB or
	CE = NM or
	CE = NS or
	CE = NZ
		CA = NB

E = NZ	CE = PZ	CA = NS

E = NZ	CE = PS or
	CE = PM or
	CE = PB or
	CE = NB
		CA = NM

E = PZ	CE = M or
	CE = N
		CA = PM

E = PZ	CE = NZ	CA = NZ

E = PZ	CE = PZ
	CE = PS or
	CE = PM or
	CE = PB
		= PB

E = PS	CE = NB or
	CE = NM or
	CE = NS or
		CA

E = PS	CE = NZ	CA Z

E = PS	CE = PZ or
	CE = PS or
	CE = PM or
	CE = PB	

E = PM	CE = NB CE = CE C	
E = PM		
E		

Figure 7. Control Rules for Example 2

N93
29567
UNCLAS

Possibilistic Measurement and Set Statistics*

Cliff Joslyn [†] [‡]

Abstract

Set-based statistics are necessary to generate possibility distributions from measured data. Methods by which physical measurements can generate statistical data on real intervals are considered, including trials from multiple heterogeneous measurement devices rather than a single instrument at multiple times; classes of consistent intervals constructed from statistical data around a common point focus or interval core; and consonant intervals constructed from statistical data.

1 Introduction

My overall interest is to expand the applications of possibility theory beyond its traditional uses in the engineering of human-created technological systems (e.g. knowledge-based control systems, artificial intelligence and approximate reasoning, etc.) to include the modeling of natural, complex systems. In order to do this, it is necessary to extend the semantics of possibility beyond traditional interpretations based on the uncertainty judgments of human subjects. Instead, a semantics of possibility that has meaning with respect to natural systems is needed.

Existing empirical methods for deriving possibility distributions are frequency conversion methods, which transform some measured probabilistic data into a possibilistic form [16]. Of course such transformations must be used when *only* frequency data are available, but the resulting possibilistic representation is never ultimately *appropriate* for data initially governed by a frequency distribution. When possibilistic data are desired, it is always preferable to obtain them in a form more directly similar to their possibilistic representation.

The additivity of frequency data results from the specificity of observations of singletons, or indeed elements of any disjoint class. Therefore, the first step towards possibilistic measurement is allowing for the possibility of non-specific measurements, that is observations that are possibly non-disjoint. This is essentially the concept of set statistics, originally advanced by Wang and Liu [17], and developed more by Dubois and Prade [4, 6].

Frequency counts on subsets result in empirically derived random sets. In earlier papers, Joslyn [9, 10] and Joslyn and Klir [11] considered methods for deriving a possibility distribution from a given empirical random set. In this paper, methods for the collection of set statistics are developed, including direct collection of interval data, and also generation of intervals from point-data streams.

2 Mathematical Preliminaries

We begin with the standard evidence and possibility theory [3, 14]. Given a finite universe $\Omega = \{\omega_i\}, 1 \leq i \leq n$, the set function $m: 2^\Omega \mapsto [0,1]$ is an **evidence function** (otherwise known as a **basic assignment** or **basic probability assignment**) when $m(\emptyset) = 0$ and $\sum_{A \subseteq \Omega} m(A) = 1$. Denote a random set generated from an evidence function as $\mathcal{S} = \{\langle A_j, m_j \rangle : m_j > 0\}$, where $\langle \cdot \rangle$ is a vector, $A_j \subset \Omega, m_j = m(A_j)$, and $1 \leq j \leq N = |\mathcal{S}| \leq 2^n - 1$. Denote the **focal set** as $\mathcal{F} = \{A_j : m_j > 0\}$ with core $\mathbf{C}(\mathcal{F}) = \bigcap_{\mathcal{F}} A_j$. The

* Prepared for the Conference of the North American Fuzzy Information Processing Society, December 1992, Puerto Vallarta.

[†] Graduate Fellow, Systems Science, SUNY-Binghamton, 327 Spring St. # 2, Portland ME, 04102, USA, (207) 774-0029, cjoslyn@bingsuns.cc.binghamton.edu, joslyn@kong.gsfc.nasa.gov.

[‡] Supported under NASA Grant # NGT 50757.

dual belief and plausibility measures on $\forall A \subset \Omega$ are $\mathrm{Bel}(A) = \sum_{A_j \subset A} m_j$ and $\mathrm{Pl}(A) = \sum_{A_j \not\perp A} m_j$, where $A \perp B \triangleq A \cap B = \emptyset$.

The **plausibility assignment** (otherwise known as the **contour function, falling shadow,** or **one-point coverage function**) of $\mathcal{S}$ is

$$\vec{\mathrm{Pl}} = \langle \mathrm{Pl}(\{\omega_i\}) \rangle = \langle \mathrm{Pl}_i \rangle, \qquad \mathrm{Pl}_i = \sum_{A_j \ni \omega_i} m_j.$$

$\vec{\mathrm{Pl}}$ is a fuzzy set that can be mapped to an equivalence class of random sets [8].

When $\forall A_j \in \mathcal{F}, |A_j| = 1$, then $\mathcal{S}$ is **specific**, and $\mathrm{Bel}(A_j) = \mathrm{Pl}(A_j) = \mathrm{Pr}(A_j)$ is an additive **probability measure** with **probability distribution** $\vec{\mathrm{Pl}} = \vec{p} = \langle p_i \rangle$ with additive normalization $\sum_i p_i = 1$. $\mathcal{S}$ is **consonant** ($\mathcal{F}$ is a **nest**) when (without loss of generality for ordering, and letting $A_0 = \emptyset$) $A_{j-1} \subset A_j$. Now $\mathrm{Pl}(A_j) = \Pi(A_j)$ is a **possibility measure**. As Pr is additive, so Π is **maximal** in the sense that $\Pi\left(\bigcup_j A_j\right) = \bigvee_j \Pi(A_j)$, where $\vee$ is the maximum operator. Denoting $A_i = \{\omega_1, \omega_2, \ldots, \omega_i\}$, and assuming that $\mathcal{F}$ is complete (i.e. $\forall \omega_i \in \Omega, \exists A_i$), then $\vec{\mathrm{Pl}} = \vec{\pi} = \langle \pi_i \rangle$ is a **possibility distribution** with maximal normalization $\bigvee_i \pi_i = 1$.

2.1 Consistency and Consonance

$\mathcal{S}$ is **consistent** when $\mathrm{C}(\mathcal{F}) \neq \emptyset$. Each consonant random set is consistent with core $\mathrm{C}(\mathcal{F}) = A_1$, and $\mathcal{F}$ being consistent is both necessary and sufficient for $\bigvee \mathrm{Pl}_i = 1$. Thus a consistent but non-consonant random set has a maximal possibility distribution $\vec{\mathrm{Pl}} = \vec{\pi}$, but its plausibility measure Pl is *not* a possibility measure Π. While an additive probability distribution uniquely determines a measure and random set, a maximal possibility distribution does not. However, a possibility measure Π^* that is optimally approximate can be constructed according to the formula $\forall A \subset \Omega, \Pi^*(A) = \bigvee_{\omega_i \in A} \pi_i$ [5]. When $\mathcal{S}$ is already consonant, then of course $\Pi^* = \mathrm{Pl} = \Pi$.

Dubois and Prade [3] suggest that the plausibility assignment of a consistent but non-consonant random set $\vec{\mathrm{Pl}} = \vec{\pi}$ should not be taken as a possibility distribution, but rather should be used to derive a nest from which a possibility distribution can be generated. That nest is the focal set of the constructed possibility measure Π^*, denoted $\mathcal{F}^* = \{B_k^*\}$. The evidence for each focal element, denoted $m_k^* = m(B_k^*)$, is given by the formula

$$m_k^* = \sum_{A_j \subset B_k} m_j - m_{k-1}^*$$

where $m_0^* = 0$. This method results in a greater constraint on the evidence provided by m, and thus the loss of some information available in a consistent $\mathcal{S}$ (see example in Section 4).

2.2 Consistent Transformations

When $\mathcal{F}$ is not consistent, then $\bigvee \mathrm{Pl}_i < 1$. Here a set of **focused consistent transformations** $\hat{\mathcal{S}}_i$ can be constructed from $\mathcal{S}$ [10, 11]. $\forall \omega_i \in \Omega$, $\hat{\mathcal{S}}_i$ is a consistent approximation of $\mathcal{S}$ with evidence function [10]

$$\hat{m}^i(A) = \begin{cases} m(A) + m(A - \{\omega_i\}), & \omega_i \in A \\ 0, & \omega_i \notin A \end{cases}.$$

The effect is to create a core $\mathrm{C}(\hat{\mathcal{F}}_i) = \{\omega_i\}$ with **focus** $\omega_i = \omega^*$. Under the transformation $\hat{\mathcal{S}}_i$, the sub-maximal plausibility assignment $\vec{\mathrm{Pl}} = \langle \mathrm{Pl}_1, \mathrm{Pl}_2, \ldots, \mathrm{Pl}_i, \ldots, \mathrm{Pl}_n \rangle$ is transformed into a maximal possibility distribution $\vec{\pi} = \langle \mathrm{Pl}_1, \mathrm{Pl}_2, \ldots, 1, \ldots, \mathrm{Pl}_n \rangle$. $\hat{\mathcal{S}}_i$ in turn generates a consonant random set $\hat{\mathcal{S}}_i^{\pi}$, determined from the constructed possibility measure Π^* of $\vec{\pi}$.

In using the transformation the task is to choose the "correct" ω^* as a focus, and to elevate the plausibility of that element to 1 as a possibilistic normalization. While there are many methods to choose ω^*, to date only the Principle of Minimal Information Distortion [10] (or Information Loss [11]) has been studied. Given a random set $\mathcal{S}$, then that focused consistent transformation $\hat{\mathcal{S}}_i$ is selected so that the total information content of $\hat{\mathcal{S}}_i^{\pi}$ is as close as possible to that of the original $\mathcal{S}$. Details of the measure of total information can be found elsewhere [7, 10, 15].

3 Empirical Random Sets

Assume that some phenomenal system can be described as a set $\Omega = \{\omega_i\}, 1 \le i \le n$. A traditional conception of a measurement on Ω results in the observation of an element $\omega_i \in \Omega$. For example, a thermometer calibrated in integral degrees on the interval $[0, 100]$ could yield a result of 72 degrees, $72 \in \{0, 1, \ldots, 100\}$.

Assume a counting function $c \colon \Omega \mapsto \mathcal{I}, c_i = c(\omega_i)$, where c_i is the number of observations of ω_i. Then for a total number of counts as N, the frequency distribution on Ω is $f \colon \Omega \mapsto [0,1], f(\omega_i) = f_i = c_i/N$. Since $\sum_i f_i = 1$, therefore f is a natural probability distribution on Ω with an additive measure $F \colon 2^{\Omega} \mapsto [0,1], F(A) = \sum_{\omega_i \in A} f_i$.

3.1 General Measuring Devices

However, most real measuring devices are not like this, due to necessary measurement uncertainty. Most measurements produce an observation of some *subset* $A \subset \Omega$, perhaps an interval $A \subset \Re$. The observation of the interval A leaves uncertainty as to the "actual" value $\omega \in A$.

It may be that not all subsets are observable. Thus a **general measuring device** is defined as a class $\mathcal{C} = \{A_{j'}\} \subset 2^{\Omega}, 1 \le j' \le N'$. The nature of the measuring device will depend on the elements and structure of $\mathcal{C}$.

Assume a collection of set observations $A^k \in \mathcal{C}, 1 \le k \le M$. In general, for some k_1, k_2, it may be that $A^{k_1} = A^{k_2}$. Therefore the A^k form a multi-set, denoted as a vector $\vec{A} = \langle A^1, A^2, \ldots, A^M \rangle$. The **empirically derived focal set** $\mathcal{F}^E \subset \mathcal{C}$ is the set of subsets that are actually observed in $\vec{A}$. $\mathcal{F}^E$ is derived by eliminating the duplicates in $\vec{A}$. Let $\mathcal{F}^E = \{A_j\}$, where $\mathcal{F}^E \subset \mathcal{C}, 1 \le j \le N \le N', N \le M$ and $\forall A_j \in \mathcal{F}^E, A_j \in \vec{A}$, and inclusion of an element in a vector is defined as would be expected.

Now establish a set-counting function $C \colon \mathcal{F}^E \mapsto \mathcal{I}, C_j = C(A_j)$, where $\forall A_j \in \mathcal{F}^E, C_j$ is the number of occurrences of A_j in $\vec{A}$. Finally the **set-frequency function** is arrived at

$$m^E \colon \mathcal{F}^E \mapsto [0,1], \qquad m^E(A_j) = m_j^E = \frac{C_j}{\sum_{A_j \in \mathcal{F}^E} C_j} = C_j/M.$$

The intention is obvious: since $\sum_j m_j^E = 1$ and $\emptyset \notin \mathcal{F}^E$, therefore m^E is a natural evidence function on Ω generating an **empirically derived random set** denoted $\mathcal{S}^E$.

3.2 Disjoint Measuring Devices

Generally, scientists strive to construct measuring devices for which $\mathcal{C}$ is disjoint; that is, $\forall A_1, A_2 \in \mathcal{C}, A_1 \perp A_2$. In such classical measuring devices, $\mathcal{C}$ is an equivalence class on Ω, yielding an observation of an $A^k \in \mathcal{C}$ unambiguous.

Virtually all traditional measuring devices are of this type. A typical example could be a thermometer, where $\Omega \subset \Re$ is some distance along a glass tube marked at certain points, say d_j, with a certain number of degrees. The A could then be the disjoint, equal length, half-open intervals $A_j = [d_j, d_{j+1})$. Observation of a specific position of the mercury (an $\omega \in A_j$) yields a specific A_j reading for the temperature. The size of the A_j relative to the size of the tube indicates the precision of the thermometer. While any particular interval A_j is usually identified with one degree reading (either d_j or d_{j+1}), it must always be kept in mind that it in fact indicates the *entire* interval $[d_j, d_{j+1})$.

Because the A_j are disjoint, observation of any one particular interval admits to no uncertainty *at the level of description of* $\mathcal{C}$. Thus in this case $\mathcal{C}$ itself can be considered as a new universe of discourse $\Omega' = \mathcal{C} = \{A_j\}$. Because the A_j are disjoint, so will the A^k.

Now m^E is the frequency of the disjoint A_j, and is thus a true probability distribution, and not an evidence function proper. Measurements from a classical measuring device are usually parameterized in time k, yielding the observations A^k as time-series point data. An additive distribution and measure are derived as for frequencies above

$$c' \colon \Omega' \mapsto \mathcal{I}, \qquad f' \colon \Omega' \mapsto [0,1], \qquad F' \colon 2^{\Omega'} \mapsto [0,1]$$

$$c'(A_j) = c'_j = C(A_j), \qquad f'(A_j) = f'_j = m_j^E, \qquad \sum_j f'_j = 1, \qquad F'(B \subset \Omega') = \sum_{A_j \in B} f'_j.$$

4 Instrument Ensembles

One way to generate measurements of intersecting subsets is to use an ensemble of classical instruments. That ensemble can be considered as either multiple, heterogeneous instruments taking separate measurements at the same time, or as a single instrument which is changing its structure over time.

Let $C^k = \left\{ A^k_{j'_k} \right\}, 1 \leq j'_k \leq N'_k = |C_k|$ be disjoint classes on Ω, and $\mathbf{F} = \{C^k\}$ be the family of such classes, $1 \leq k \leq M$. The natural partial order on $\mathbf{F}$ is

$$C^1 \prec C^2 \quad \triangleq \quad \forall A^2_{j_2} \in C^2, \quad \exists \{A^1_{j_1}\} \subset C^1, \quad A^2_{j_2} = \bigcup A^1_{j_1}.$$

When $C^1 \prec C^2$ then C^1 refines C^2, and C^2 coarsens C^1. For example, C^1 could be a thermometer reading in tenths of degrees, while C^2 could belong to a mutually calibrated thermometer reading in whole degrees. $\mathbf{F}$ is **consonant** whenever the C^k are all comparable under $\prec$ (they are all mutual refinements or coarsenings).

Leting A^k be the subset observed in device C^k, then the vector of observations over $\mathbf{F}$ is $\vec{A} = \langle A^k \rangle, |\vec{A}| = M$, and $\vec{A}$ generates the empirical random set $\mathcal{S}^E$ as described in Section 3.1. If any of the C^k share common members (in particular, if any of them are equal), then some of the A^k may be equal, yielding multiple observations in $\vec{A}$ of certain subsets. Otherwise, all subsets will be observed a single time, and will not necessarily be disjoint.

Assume observations from two devices, say $A^1 \in C^1$ and $A^2 \in C^2$. It is expected that $A^1 \not\perp A^2$. In the event that $A^1 \perp A^2$, then at least one of the devices C^1 or C^2 would be regarded as being in error, or perhaps even the assumption of the "reality" of the quantity being measured would be questioned. Thus, while there is nothing in the mathematics that would preclude such a result, pragmatic conditions require that $\mathcal{F}^E$ be consistent, so that $\mathcal{S}^E$ has a natural possibility distribution π and at worst a constructed possibility measure Π^*. In the event that $\mathcal{F}^E$ is nevertheless not consistent, and there are pragmatic reasons for accepting the results of the measurement, then the focused consistent transformation method outlined in Section 2.2 is available to construct consistent random sets $\hat{\mathcal{S}}_i$.

When $\mathbf{F}$ is consonant, then without loss of generality for ordering, $C^1 \prec C^2 \prec \ldots \prec C^M$. Here if $\mathcal{F}^E$ is consistent, then it must also be consonant, with $A_1 \subset A_2 \subset \ldots \subset A_N$. Of course, in this case a possibilistic analysis is less useful than it would be otherwise, since there is an absolute gain in accuracy in the movement towards the finest measurement A^1. Nevertheless, the mathematical analysis is available.

Example 1: Let $\Omega = [0, 5] \subset \Re$ and define a family $\mathbf{F}$ of four measuring devices

$$C^1 = \{[0,1),[1,2),[2,3),[3,4),[4,5]\}, \qquad C^2 = \{[0,1),[1,2),[2,3.5),[3.5,5]\},$$
$$C^3 = \{[0,1.5),[1.5,3.5),[3.5,4),[4,5]\}, \qquad C^4 = \{[0,1.5),[1.5,4),[4,5]\},$$

so that $M = 4$. $\mathbf{F}$ is not consonant, but $C^3 \prec C^4$. Measurements are made on each instrument yielding a vector of four measurements (Figure 1)

$$\vec{A} = \langle [1,2),[1,2),[1.5,3.5),[1.5,4) \rangle.$$

After eliminating duplicates, the set of observed intervals $\mathcal{F}^E$ is derived with $N = 3 < M$ and

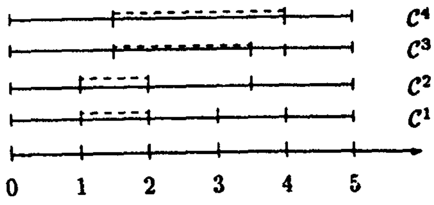


Figure 1: Measurements on four instruments.

random set $\mathcal{S}^E$

$$\mathcal{F}^E = \{[1,2),[1.5,3.5),[1.5,4)\}, \qquad \mathcal{S}^E = \{\langle [1,2),.5 \rangle, \langle [1.5,3.5),.25 \rangle, \langle [1.5,4),.25 \rangle\}.$$

$\mathcal{F}^E$ is consistent with core $C(\mathcal{F}^E) = [1.5, 2)$, the region on which $\pi = 1$. $\pi(\omega)$ is determined by $\pi(\omega) = \sum_{A_j \ni \omega} m_j^E$, so that

$$\pi(\omega) = \begin{cases} .5, & \omega \in [1, 1.5) \\ 1, & \omega \in [1.5, 2) \\ .5, & \omega \in [2, 3.5) \\ .25, & \omega \in [3.5, 4) \\ 0, & \text{elsewhere} \end{cases}$$

as shown in Figure 2.

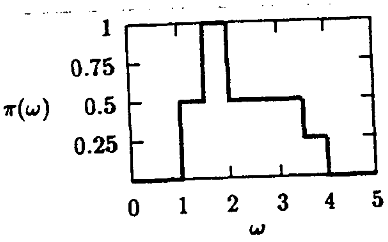


Figure 2: π determined from $\mathcal{S}^E$.

Dubois and Prade's method described in Section 2.1 results in the consonant random set

$$\{\langle [1.5, 2), 0\rangle, \langle [1, 3.5), .75\rangle, \langle [1, 4), .25\rangle\}$$

and possibility distribution shown in Figure 3. Comparing Figures 2 and 3, it can be seen that reliance on the consonant class and its greater constraint results in a loss of distinctions of possibility values over portions of the possibility curve.

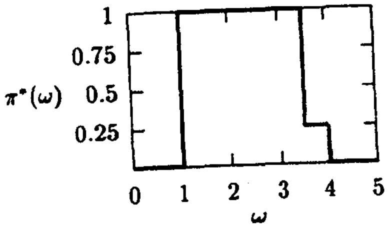


Figure 3: π^* determined from Dubois and Prade's method.

Because $\mathcal{F}^E$ is finite, π is piecewise continuous, consisting of a union of constant segments. Also, because $C(\mathcal{F}^E)$ is connected, π is unimodal at C. Therefore π in this example, and in the sections to follow, has the form of a centrally peaked staircase. As $|\mathcal{F}^E| \longrightarrow \infty$, π approaches the traditional forms for possibility distributions (e.g., fuzzy numbers [2]).

5 Consistent Intervals from Focused Point Data

Even given a single measuring device and time-series data gathered on it (as discussed in Section 3.2), which is our normal concept of measurement, interval data can still be generated. Since classical instruments generate observations of disjoint intervals that can be regarded as distinct points in a higher-level state space, therefore in the following sections a single measuring device that yields observations of points in a lower-level state space, a closed interval $\Omega \subset \Re$, will be considered.

Denote an observation as a data point $d \in \Omega$, and the collection of data as a **data stream**, a multiset denoted as the vector $\vec{D} = \langle d_i \rangle, 1 \leq i \leq n$. The set generated by eliminating duplicates in $\vec{D}$ is the **data set** $D = \{d_i'\}, 1 \leq i \leq n' \leq n$.

A possibilistic analysis of $\vec{D}$ will be approached by using its **order statistics** [1]. For a given data stream $\vec{D}$, the order statistics, denoted $d_{(i)}$, are a permutation of the d_i such that $d_{(1)} \leq d_{(2)} \leq \ldots \leq d_{(n)}$. $d_{(1)}$ and $d_{(n)}$ are called the **extremes**, and the **range** interval is $W = [d_{(1)}, d_{(n)}]$. The order statistics of the data set D' are $d'_{(i)}, 1 \leq i \leq n'$. The $d'_{(i)}$ naturally generate the disjoint intervals $\delta_i = \left[d'_{(i)}, d'_{(i+1)}\right), 1 \leq i \leq n' - 1$. For completeness, let $\delta_{n'} = \left[d'_{(n')}, d'_{(n')}\right] = \left\{d'_{(n')}\right\}$. Let the set of disjoint intervals be $\Delta = \{\delta_i\}$, so that $\bigcup_{\delta_i \in \Delta} \delta_i = \Omega$.

5.1 Focused Data Intervals

Δ thus represents a classical measuring device with the δ_i partitioning W, and so the greatest problem with deriving a possibility distribution from Δ is the lack of a focus, or any core. Thus we posit the existence of a focus $u \in W$. The purpose of u is to provide a value on which all the intervals (yet to be determined) agree; a value for which $\pi(u) = 1$. u naturally divides W into left and right sub-intervals denoted $W_l = [d_{(1)}, u)$ and $W_r = (u, d_{(n)}]$ so that $W_l \cup [u,u] \cup W_r = W$.

Given a focus $u \in W$, then $\forall d_{(i)} \neq u, d_{(i)} \in W_l$ or $d_{(i)} \in W_r$. Denote the intervals $A^i, 1 \leq i \leq n$ as follows:

$$A^i = \begin{cases} [d_{(i)}, u], & d_{(i)} \in W_l \\ [u, d_{(i)}], & d_{(i)} \in W_r \\ [u,u], & d_{(i)} = u \end{cases}.$$

Since

$$d_{(i_1)}, d_{(i_2)} \in W_l, \quad i_1 \leq i_2 \quad \rightarrow \quad A^{i_2} \subset A^{i_1}; \qquad \text{and} \qquad d_{(i_1)}, d_{(i_2)} \in W_r, \quad i_1 \leq i_2 \quad \rightarrow \quad A^{i_1} \subset A^{i_2},$$

therefore each of the sets of intervals

$$\mathcal{F}_l = \{A^i : d_{(i)} \in W_l\}, \qquad \mathcal{F}_r = \{A^i : d_{(i)} \in W_r\},$$

are nests. Since $\forall i, u \in A^i$, the total set $\{A^i\}$ is consistent, forming a focal set $\mathcal{F}^E = \mathcal{F}_l \cup \mathcal{F}_r$ with core $\mathbf{C}(\mathcal{F}^E) = [u,u] = \{u\}$. $\mathcal{S}^E$ is then constructed from the counts of the $d_{(i)} \in \vec{D}$ of the corresponding interval A^i.

Generally, each $d_{(i)}$ will generate a single count for the interval A^i. However, if $\exists i_1, i_2, A^{i_1} = A^{i_2}$ then multiple counts will be generated as discussed in Section 4. If $u = d_{(1)}$ or $u = d_{(n)}$ then $\mathcal{F}^E$ will actually be consonant.

Example 2: As above, let $\Omega = [0,5]$, and assume that $n = 6$ point observations in Ω are taken giving the data stream $\vec{D} = \langle 2, 1, 4, 1.5, 2, 4.5 \rangle$. The order statistics are

$$d_{(1)} = 1, \quad d_{(2)} = 1.5, \quad d_{(3)} = d_{(4)} = 2, \quad d_{(5)} = 4, \quad d_{(6)} = 4.5$$

and $W = [1, 4.5]$. The corresponding data set is $D' = \{1, 1.5, 2, 4, 4.5\}$ so that $n' = 5 < n$, with order statistics and disjoint intervals

$$d'_{(1)} = 1, \quad d'_{(2)} = 1.5, \quad d'_{(3)} = 2, \quad d'_{(4)} = 4, \quad d'_{(5)} = 4.5$$

$$\Delta = \{[1, 1.5), [1.5, 2), [2, 4), [4, 4.5), [4.5, 4.5]\}$$

Assuming that $u \in [2,4]$, then the focal and random sets (Figure 4, with $u = 3$) are

$$\mathcal{F}^E = \mathcal{F}_l \cup \mathcal{F}_r = \{[1,u], [1.5,u], [2,u]\} \cup \{[u,4], [u,4.5]\},$$

$$\mathcal{S}^E = \{\langle [1,u], 1/6 \rangle, \langle [1.5,u], 1/6 \rangle, \langle [2,u], 1/3 \rangle, \langle [u,4], 1/6 \rangle, \langle [u,4.5], 1/6 \rangle\}.$$

The possibility distribution is shown in Figure 5.

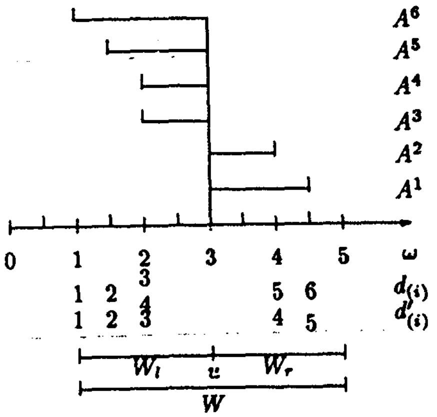


Figure 4: Consistent family from focused data set.

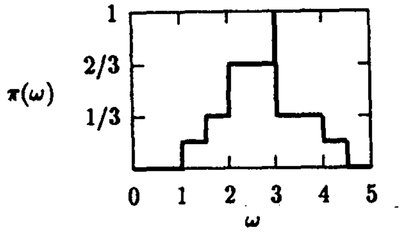


Figure 5: Derived possibility distribution.

5.2 Choice of Focus

So far the method by which the focus u can be chosen has not been discussed. While a number of methods suggest themselves, selection of methods will depend on user methodology and further empirical research. However, in Example 2 the first four methods below all yield $u \in [2, 4]$, which is the inner interval of Δ (see Section 6).

Sample Mean: Selection of

$$u = \bar{D} = \sum d_i / n$$

is a possibility, although one that is not in keeping with possibilistic concepts. In our example, this would yield $u = 2.5$.

Range Midpoint: The midpoint of W, denoted $\bar{W}$, is much more in keeping with possibilistic concepts:

$$u = \bar{W} = \frac{d_{(1)} + d_{(n)}}{2}.$$

It expresses something like the concept of a "possibilistic sample mean". This would yield $u = 2.75$ in the example.

Closest to Range Midpoint: There may be some value in having u actually be one of the data points, so that $u \in D'$. This can be done by selecting that $d'_i \in D'$ closest to $\bar{W}$ (yielding $u = 2$ in our example):

$$u = \min_{d'_i \in D'} |d'_i - \bar{W}|.$$

Data-Set Midpoint: The middle point of the data set itself can be chosen, that is

$$u = d'_{\left(\frac{n'+1}{2}\right)}$$

if n' is odd. If n' is even, then either

$$u = d'_{(n'/2)} \text{ or } u = d'_{\left(\frac{n'}{2}+1\right)}.$$

Alternatively, if n' is even then the midpoint of the central interval can be selected:

$$u = \frac{d'_{(n'/2)} + d'_{\left(\frac{n'}{2}+1\right)}}{2}.$$

Information Principles: Finally, the Information Principles introduced in Section 2.2 can be applied to the problem [11, 12]. Again, details will not be given here. Selection of u can be regarded as a problem of ampliative reasoning, of making an inductive inference beyond the given information. Then the Principle of Maximum Uncertainty can be invoked, which states that u should be chosen so as to maximize the total uncertainty of the resulting random set, or of the final possibility distribution.

Alternatively, selection of u can be regarded as one of transformation from the frequency distribution of $\vec{D}$ to a possibility distribution. Then the Principle of Uncertainty Invariance [13] or Minimal Information Distortion [10] can be used, which states that u should be chosen so as to make the total uncertainty of $\mathcal{S}^E$ as close as possible to the entropy $\vec{D}$.

6 Interval Cores

A potential disadvantage of the methods in Section 5 is the reliance on a singleton-valued core set $C(\mathcal{F}^E) = \{u\}$, while the other elements of the method are the intervals δ_i and A^i. Instead, methods that yield an interval-valued core can be considered. A disadvantage of these methods is that they may eliminate some data points, thus loosing some information from the resulting $\mathcal{S}^E$.

Let $\vec{D}, D'$ and Δ be given as above. Now identify the core as an interval in the range with endpoints C_l and C_r, so that $C = [C_l, C_r] \subset W$. Assume for the moment that $\nexists d_{(i)} \in C$. Then the left and right ranges can be redefined as $W_l = [d_{(1)}, C_l)$ and $W_r = (C_r, d_{(n)}]$, so that $W_l \cup C \cup W_r = W$. Also redefine the intervals A^i as follows:

$$A^i = \begin{cases} [d_{(i)}, C_r], & d_{(i)} \in W_l \\ [C_l, d_{(i)}], & d_{(i)} \in W_r \end{cases}.$$

Again $\mathcal{F}_l$ and $\mathcal{F}_r$ are nests, so that $\mathcal{F}^E = \mathcal{F}_l \cup \mathcal{F}_r$ is consistent with core $C(\mathcal{F}^E) = \bigcap A^i = C$.

If $\exists \{d_{(k)}\} \subset C$, then a new data set $\vec{D}^- = \vec{D} - \{d_{(k)}\}$ is defined, where the operation $-$ of a set from a vector is just the elimination of $\forall d_{(k)}$ from $\vec{D}$. Corresponding new $d^-_{(i)}, D'^-$, etc. can be generated without special treatment.

6.1 Choice of C

As with the selection of point foci, there are a variety of methods by which an interval core can be selected.

Central Disjoint Interval: If n' is even, then a central disjoint interval is naturally generated from the data set D':

$$C = \delta_{n'/2}.$$

Since $d'_{(n'/2)}, d'_{\left(\frac{n'}{2}+1\right)} \in C$, all instances of them will be eliminated from $\vec{D}$ in forming $\vec{D}^-$.

Modified Central Interval: If n' is odd, then there are two disjoint intervals on either side of $d'_{\left(\frac{n'+1}{2}\right)}$. Thus a core would be selected

$$C = \delta_{\frac{n-1}{2}} \cup \delta_{\frac{n+1}{2}},$$

that eliminates instances of the three data points $d'_{(\frac{n'-1}{2})}, d'_{(\frac{n'+1}{2})}$, and $d'_{(\frac{n'+3}{2})}$ from $\vec{D}$.

Alternatively, the midpoints of the two disjoint intervals around $d'_{(\frac{n'+1}{2})}$ can be selected as the endpoints of $\mathbf{C}$:

$$\mathbf{C} = \left[\frac{d'_{(\frac{n'-1}{2})} + d'_{(\frac{n'+1}{2})}}{2}, \frac{d'_{(\frac{n'+1}{2})} + d'_{(\frac{n'+3}{2})}}{2} \right].$$

Disjoint Interval Around Focus: Given a method from Section 5.2 to select a point focus u, then $\mathbf{C}$ can just be selected as the data-generated disjoint interval around u:

$$\mathbf{C} = \delta_i, \quad u \in \delta_i.$$

As above, instances of $d'_{(i)}$ and $d'_{(i+1)}$ will be eliminated from $\vec{D}$.

Confidence Interval Around Focus: It may be appropriate for the user to involve some traditional statistical information. Again, given some focus u, then $\mathbf{C}$ can be selected as the interval within a standard deviation of u:

$$\mathbf{C} = \left[u - \sigma(\vec{D}), u + \sigma(\vec{D}) \right].$$

Information Principles: Methods of Uncertainty Maximization or Invariance can be applied, as discussed in Section 5.2.

7 Consonant Intervals from Focused Point Data

It may be desirable to go as far as generating consonant, not just consistent, families from a data stream $\vec{D}$. However, as the methods progress from consistent families with point focuses, through consistent families with interval cores, to consonant classes, the constraint on $\mathcal{S}^E$ increases, thus loosing information available in the original $\vec{D}$. This is reflected in the loss of some data points in the interval core methods, and in roughly half the number of available intervals from the following consonant methods. Thus as with the case of an ensemble of measuring devices (Section 4), use of strictly consonant cases may be less useful than simply consistent cases.

Again, a number of methods present themselves.

Inner Nested Intervals from Interval Core: Assume that an interval core $\mathbf{C} = [\mathbf{C}_l, \mathbf{C}_r]$ has been determined according to some method discussed in Section 6.1. Denote $A^1 = \mathbf{C}$, and construct a set of intervals $A^k = [A^k_l, A^k_r]$ such that $A^k_l, A^k_r \in D'$ and $A^k \subset A^{k+1}$. Given an interval A^k, then A^{k+1} is the nearest interval determined by D' containing A^k

$$A^{k+1}_l = \max_{d'_{(i)} \in D'} d'_{(i)} < A^k_l, \qquad A^{k+1}_r = \min_{d'_{(i)} \in D'} d'_{(i)} > A^k_r.$$

The A^k are available up to a maximal $A^{\lfloor n'/2 \rfloor} = W$. $\mathcal{F}^E = \{A^k\}$ is then a consonant class. The count of A^k can be determined as the maximum number of occurrences of either endpoint of A^k in $\vec{D}$.

Inner Nested Intervals from Point Focus: Assume instead that a point core $u \in W$ has been determined according to some method discussed in Section 5.2. Now simply let $A^1 = [u, u]$ and apply the method above.

Outer Nested Intervals: Proceed in the opposite direction from above. Now define $A^1 = W$, and construct A^{k+1} from A^k as follows:

$$A^{k+1}_l = \min_{d'_{(i)} \in D'} d'_{(i)} > A^k_l, \qquad A^{k+1}_r = \max_{d'_{(i)} \in D'} d'_{(i)} < A^k_r.$$

Acknowledgements

Prof. George Klir and Dr. Peter Cariani provided useful discussion on these subjects. I would also like to thank an anonymous reviewer for his or her helpful comments.

References

[1] David, Herbert A: (1981) *Order Statistics*, Wiley, New York

[2] Dubois, Didier: (1987) "Fuzzy Numbers: An Overview", in: ***Analysis of Fuzzy Information***, v. **1** of **3**, pp. 3-39, CRC Press, Boca Raton

[3] Dubois, Didier and Prade, Henri: (1988) ***Possibility Theory***, Plenum Press, New York

[4] Dubois, Didier and Prade, Henri: (1989) "Fuzzy Sets, Probability, and Measurement", *Eur. J. of Operational Research*, v. **40**:2, pp. 135-154

[5] Dubois, Didier and Prade, Henri: (1990) "Consonant Approximations of Belief Functions", *Int. J. Approximate Reasoning*, v. **4**, pp. 419-449

[6] Dubois, Didier and Prade, Henri: (1992) "Evidence, Knowledge and Belief Functions", *Int. J. Approximate Reasoning*, v. **6**:3, pp. 295-320

[7] Geer, James and Klir, George: (1991) "Discord in Possibility Theory", *Int. J. Gen. Sys.*, v. **19**, pp. 119-132

[8] Goodman, IR and Nguyen, HT: (1986) ***Uncertainty Models for Knowledge-Based Systems***, North-Holland, Amsterdam

[9] Joslyn, Cliff: (1991) "Towards an Empirical Semantics of Possibility Through Maximum Uncertainty", in: *Proc. IFSA 1991*, v. **A**, pp. 86-89

[10] Joslyn, Cliff: (1992) "Empirical Possibility and Minimal Information Distortion", in: *Fuzzy Logic: State of the Art*, ed. R. Lowen, Kluwer, in press

[11] Joslyn, Cliff and Klir, George: (1992) "Minimal Information Loss Possibilistic Approximations of Random Sets", in: *Proc. 1992 FUZZ-IEEE Conference*, ed. Jim Bezdek, pp. 1081-1088, IEEE, San Diego

[12] Klir, George: (1990) "Uncertainty Principles in Reasoning", in: *Proc. 1990 NAFIPS*, pp. 212-214

[13] Klir, George: (1990) "Principle of Uncertainty and Information Invariance", *Int. J. Gen. Sys.*, v. **17**:2-3, pp. 249-275

[14] Klir, George and Folger, Tina: (1987) ***Fuzzy Sets, Uncertainty, and Information***, Prentice Hall

[15] Klir, George and Parviz, Behvad: (1992) "A Note on the Measure of Discord", in: *Proc. 8th Conf. on Uncertainty in AI*, Stanford

[16] Klir, George and Parviz, Behvad: (1992) "Possibility-Probability Conversions: An Empirical Study", in: *Progress in Cybernetics and Systems*, ed. R. Trappl, pp. 19-26, Hemisphere, New York

[17] Wang, PZ and Liu, XH: (1984) "Set-Valued Statistics", *J. Eng. Math.*, v. **1**:1, pp. 43-54

N93
29568
UNCLAS

THE FUSION OF INFORMATION VIA FUZZY INTEGRATION

James M. Keller and Hossein Tahani
Department of Electrical and Computer Engineering
University of Missouri-Columbia
Columbia, MO 65211

ABSTRACT

Multisensor fusion is becoming increasingly important in intelligent computer vision systems. In this paper we present the generalized fuzzy integral with respect to an S - decomposable measure as a tool for fusing information from multiple sensors in an object recognition problem. Results from an experiment with automatic target recognition imagery are provided.

INTRODUCTION

Many intelligent systems use multiple information sources because the information from any individual source is either partial or contaminated, that is, it is uncertain and/or imprecise. To evaluate this information properly, intelligent systems must be capable of integrating both complementary and redundant information provided by multiple knowledge sources. Pattern classifiers, scene analysis systems, image processing systems, and computer vision systems all must be capable of integrating knowledge from multiple sources.

In an earlier work, we developed a new evidence fusion technique, based on the fuzzy integral with respect to g_λ-fuzzy measures [1]. The fuzzy integral differs from the previously mentioned paradigms in that both objective evidence supplied by various sources and the expected worth of the subsets of these sources are considered in the fusion process. In [2] we developed the fuzzy integral with respect to different classes of fuzzy measures, namely, S-decomposable measures, as an information fusion technique. We generalized the concept of the fuzzy integral to increase the flexibility in the "rule of combination" of evidence. In this paper, we briefly survey that development and demonstrate the usefulness of the generalized fuzzy integral in a multisensor fusion domain.

FUZZY MEASURES

Let X be a finite set and let Ω be the power set of X. The elements of Ω are called measurable subsets of X.

Definition 1: A set function $\mu : \Omega \rightarrow [0,1]$ is called a fuzzy measure iff the following axioms hold.

(1) $\mu(\emptyset) = 0$, $\mu(X) = 1$,

(2) $\mu(A) \leq \mu(B)$ if $A \subset B$,

Fuzzy measures based on triangular cononrms (t-conorms) have been studied by Dubois and Prade in [3]. They have shown many interesting properties of these types of fuzzy measures and their relation to Shafer's belief and plausibility measures. Weber in [4] studied the fuzzy measures based on Archimedean t-conorms to define the Weber integrals. He called the fuzzy measures based on t-conorms S-decomposable measures.

We define the S-decomposable measures, following Weber. Let X be a finite set. Note that this restriction is only for simplicity and that all of our applications assume finite sets, but the theory can be extended to infinite sets (see [4]).

Definition 2: A function $\mu : \Omega \longrightarrow [0,1]$ with $\mu(\emptyset) = 0$ and $\mu(X) = 1$ is called an S - decomposable measure with respect to a t-conorm S iff for A, B $\subseteq$ X with $A \cap B = \emptyset$,

$$\mu(A \cup B) = S(\mu(A), \mu(B)).$$

Definition 3: A mapping $X \longrightarrow [0,1]$ defined by $x_i \longmapsto \mu(\{x_i\}) = \mu^i$ is called a <u>fuzzy density mapping</u> and the set $\{\mu^1, \ldots, \mu^n\}$ is called the <u>fuzzy density set</u>.

We note that an S-decomposable measure is uniquely defined by knowing the t-conorm and the fuzzy density mapping. Let $X = \{x_1, \ldots, x_n\}$ be a finite set and let $\mu^i = \mu(\{x_i\})$. If A is a subset of X, $A = \{y_1, \ldots, y_p\}$, then

$$\mu(A) = \mu(\{y_1\} \cup \ldots \cup \{y_p\}) = S\left[\mu(\{y_1\}), \ldots, \mu(\{y_p\})\right].$$

Now, since $\mu(X) = 1$, the fuzzy densities must satisfy

$$S(\mu^1, \ldots, \mu^n) = 1.$$

This equality would be trivially true if $\mu^i = 1$ for some i. Thus a S-decomposable measure can be constructed by knowing the density mapping and assuming that at least one of the fuzzy densities is 1.

THE FUZZY INTEGRAL

Definition 4: Let X be a finite set and Ω be the power set of X. Let h be a function from X into the closed interval [0,1]. The fuzzy integral over $A \subseteq X$ of the function h with respect to a fuzzy measure μ is defined by

$$\int_A h(x) \circ \mu(\cdot) = \sup_{E \subseteq X}\left[\min\left[\min_{x \in E} h(x)\ ,\ \mu(A \cap E)\right]\right].$$

The calculation of the fuzzy integral when X is a finite set is easily given. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a finite set and let $h : X \rightarrow [0,1]$ be a function. Suppose $h(x_1) \geq h(x_2) \geq \ldots \geq h(x_n)$, (if not, X is rearranged so that this relation holds). Then Sugeno in [5] proved that a fuzzy integral, e, with respect to a fuzzy measure μ over X can be computed by

$$e = \max_{i=1}^{n}\left[\min\left[h(x_i), \mu(A_i)\right]\right], \qquad (1)$$

where $A_i = \{x_1, \ldots, x_i\}$.

Equation (1) is the "rule of combination" for the fuzzy integral-based information fusion. If μ is an S-decomposable measure, then the values of $\mu(A_i)$ can be determined recursively as

$$\mu(A_1) = \mu(\{x_1\}) = \mu^1 \qquad (2a)$$

$$\mu(A_i) = S(\mu^i\ ,\ \mu(A_{i-1})), \quad 2 \leq i \leq n. \qquad (2b)$$

The reader is directed to [1,2,5] for many theoretical properties of fuzzy measures and fuzzy integrals.

Let $\{\mu^i : 1 \leq i \leq n\}$ be a fuzzy density set and let $\mu_\sigma^{\mathcal{D}}$, $\mu_\sigma^{\mathcal{B}}$, μ_σ^{Π}, and $\mu_\sigma^{\mathcal{M}}$ denote the fuzzy measures based on $S_{\mathcal{D}}$, $S_{\mathcal{B}}$, S_{Π}, and $S_{\mathcal{M}}$ t-conorms [6] (see appendix for definitions) respectively, for a permutation σ of the fuzzy density set. Since $S_{\mathcal{D}} > S_{\mathcal{B}} > S_{\Pi} > S_{\mathcal{M}}$, then for any subset A of X,

$$\mu_\sigma^{\mathcal{D}}(A) \geq \mu_\sigma^{\mathcal{B}}(A) \geq \mu_\sigma^{\Pi}(A) \geq \mu_\sigma^{\mathcal{M}}(A). \qquad (3)$$

For these measures, the following theorem (proved in [2]) is of interest.

Theorem 1: Let $\{\mu^i : 1 \leq i \leq n\}$ be a fuzzy density set that define a fuzzy measure. Let h be a function from X to [0,1]. Then

$$\int_X h(x) \circ \mu_\sigma^{\mathcal{D}}(\cdot) \geq \int_X h(x) \circ \mu_\sigma^{\mathcal{S}}(\cdot) \geq \int_X h(x) \circ \mu_\sigma^{\Pi}(\cdot) \geq \int_X h(x) \circ \mu_\sigma^{\mathcal{M}}(\cdot).$$

THE GENERALIZED FUZZY INTEGRALS

In the definition of the fuzzy integral Sugeno, in a loose sense, used the max and the min operators to replace the addition and the multiplication in the Lebesgue's integral. It seems natural to generalize the fuzzy integral by using a t-norm instead of the min operator and by replacing the max operator with a t-conorm [6].

In [2] we suggested two types of generalizations of the fuzzy integral which have natural interpretations. The fuzzy integral, as defined in equation (1), may be interpreted as "the highest pessimistic" grade of agreement between the objective evidence,h, and the expectation, μ. For the first generalization, we replace the min operator by any t-norm, ranging from $T_{\mathcal{M}}$ to $T_{\mathcal{D}}$ (see appendix). The resultant integrals can be interpreted as ranging from "the highest pessimistic" to "the lowest pessimistic" grade of agreement between h ana μ.

Let X be a finite set. Let h be function from X into the closed interval [0,1] and assume that h is sorted in decreasing order. Then the above generalization of the fuzzy integral of tha fuction h with respect to a fuzzy measure μ is written as

$$e_T = \max_{i=1}^{n} \left[T\left(h(x_i), \mu(A_i) \right) \right], \qquad (4)$$

where $A_i = \{x_1, \ldots, x_i\}$, and T is a t-norm. For example, e_Π is the integral value of the function h with respect to a fuzzy measure μ when the t-norm T_Π is used instead of the min operator, $T_{\mathcal{M}}$.

In [7] an alternative definition of the fuzzy integral of the function h with respect to a fuzzy measure μ is given by

$$\int_A h(x) \circ \mu(\cdot) = \inf_{E \subseteq X} \left[\max \left\{ \min_{x \in E} h(x) \ , \ \mu(A \cap E) \right\} \right].$$

When X is a finite set and $h(x_1) \geq \ldots \geq h(x_n)$, an optimistic version of this integral can be calculated by

$$E = \min_{i=1}^{n} \left[\max \left\{ h(x_i), \mu(A_i) \right\} \right], \qquad (5)$$

where $A_i = \{x_1, \ldots, x_i\}$. This integral can be interpreted as "the lowest optimistic" grade of agreement between h and μ. Then by replacing the max operator by any t-conorm ranging from $S_{\mathcal{M}}$ to $S_{\mathcal{D}}$, the resultant integrals will be interpreted as ranging from "the lowest optimistic" to "the highest optimistic" grade of agreement between h and μ. Similar to (4), letting E_S to denote the generalized fuzzy integral of the function h (assuming h is sorted in decreasing order) with respect to a fuzzy measure μ using the t-conorm S instead of max operator in (5), we can write

$$E_S = \min_{i=1}^{n} \left[S\left\{ h(x_i), \mu(A_i) \right\} \right], \qquad (6)$$

where $A_i = \{x_1, \ldots, x_i\}$.

The following theorems (proved in [2]) establish an ordering of the generalized fuzzy integrals for a fixed function h and fuzzy measure μ.

Theorem 2: Let μ be a fuzzy measure and $h : X \longrightarrow [0,1]$. Let $S_1 \leq S_2$ be two t-conorms. Then $E_{S_1} \leq E_{S_2}$.

Corollary 1: Let μ be a fuzzy measure and $h : X \longrightarrow [0,1]$. Then $E_{\mathcal{D}} \geq E_{\mathcal{B}} \geq E_{\Pi} \geq E_{\mathcal{M}}$.

Similar to theorem 2, we have:

Theorem 3: Let μ be a fuzzy measure and $h : X \longrightarrow [0,1]$. Let $T_1 \leq T_2$ be two t-norms. Then $e_{T_1} \leq e_{T_2}$.

Corollary 2: Let μ be a fuzzy measure and $h : X \longrightarrow [0,1]$. Then $e_{\mathcal{M}} \geq e_{\Pi} \geq e_{\mathcal{S}} \geq e_{\mathcal{D}}$.

APPLICATIONS - MULTISENSOR FUSION

The fuzzy integral was used as a segmentation tool in [8-9], and as a fusion technique in [1-2]. Here, the design and the implementation of a multisensor object recognition system using the generalized fuzzy integrals with respect to t-conorm-based fuzzy measures (S-decomposable measures) is explained.

At any level of recognition, the classification problem can be stated as follows : Let $C = \{C_1,\dots,C_m\}$ be a set of classes or hypotheses of interest. Let A be an object under consideration in the scene. Then one must decide to which class C_i, object A belongs. Note that each C_i may, in fact, be a set of classes by itself.

Let $X = \{x_1,\dots,x_n\}$ be a finite set. Each x_i is a knowledge source or may itself be a set of knowledge sources for the recognition of a particular class, C_i, $1 \leq i \leq m$. Let A be the object under consideration for recognition. Let $h_k : X \rightarrow [0,1]$ be the partial evaluation of the object A for class C_k, that is, $h_k(x_i)$ is an indication of how certain we are in classifying the object A in class C_k using the knowledge source x_i.

In order to calculate the fuzzy integral value, the degree of importance, μ_k^i, of how significant x_i is in the recognition of the class C_k, must be given. These densities can be subjectively assigned by an expert, or can be generated from a training data set, as in [1,2,9].

After sorting the h function in descending order (along with their corresponding densities), we can construct the S-decomposable measure, μ, using equations (2). Now, using equations (4) or (6), the generalized fuzzy integral value can be calculated.

```
DO FOR each object
    DO FOR each class
        Get $h_k(x_i)$
        Sort $h_k(x_i)$ in descending order
        Calculate measures recursively by equation 2
        Calculate genenralized fuzzy integrals by equation 4 or 6
    END DO
    Classify object into class with largest integral value
    END DO
```

RESULTS

The data consists of several sequences of FLIR (forward looking infrared) and and TV images containing an armored personnel carrier (APC) and two different tanks. There were five 100 frame sequences of FLIR and two 100 frame sequences of TV images.Sequence 5 of FLIR and sequence 1 of TV were taken simultaneously and constitute the multi-sensor data.

Size-contrast filters were run on each image to detect objects of interest. Several different statistical and texture features were calculated for the object windows found by the prescreeing operation. Here, the features are assumed to support the existence of an object directly. In this experiment the system was tested by multi-sensor data on sequence 5 of FLIR and sequence 1 of TV, using sequence 4 of FLIR and sequence 2 of TV for training. The h functions were generated using the (smoothed) normalized histogram of the training data, and the fuzzy densities were generated using the method described in [10]. Here, we consider the problem of target vs. non-target. In this problem there were 11 features for each sensor. These features consisted of four statistical features and seven texture features calculated on the unsegmented objects. In a [11], we subdivided this problem into specific classification problems, and investigated the effect of a multilayer structure on the multisensor fusion problem for object recognition.

Tables 1 and 2 show these results. Table 1 shows the result of using different integration values for final classification and Table 2 shows the confusion matrix of the best overall classification for the problem. As it can be seen from Table 1, the best total correct classifications occurs for $E_{\mathcal{D}}$, the highest optimistic integral value.

This is due to the fact that many sources (features) provided zero values for all classes including the correct class. The reason for this is the training data used to generate the h values was considerably different from the testing data. The testing data was registered multi-sensor data whereas the training data consisted of two non-corresponding sets of single sensor data, a practical problem when dealing with real data. The robustness of this approach is demonstrated by the fact that, except for the most pessimistic integral , over 68% of the data was correctly classified in spite of the problem with the data.

Table 1

% Total Correct Classifications for 1-level Configuration

	$E_{\mathcal{D}}$	$E_{\mathcal{S}}$	E_{Π}	$E_{\mathcal{M}}$	$e_{\mathcal{M}}$	e_{Π}	$e_{\mathcal{S}}$	$e_{\mathcal{D}}$
	96.9	78.2	80.1	79.1	68.7	71.8	75.5	9.8

Table 2
Confusion Matrix of Best Overall Classification

	Target	Non-target
Target	300	0
Non-target	10	16

CONCLUSIONS

In this paper, a generalization of an earlier methodology for information fusion using the generalized fuzzy integral with respect to a fuzzy measure based on a t-conorm was applied to the problem of multisensor fusion. S-decomposable measures allow the prediction of the effects of changes in importance of nodes to the overall evaluation. Also, these measures can simulate the different attitudes necessary for information fusion.

The generalized fuzzy integral algorithm as a multisensor fusion paradigm was applied to the problem of automatic target recognition and produced excellent results.

REFERENCES

[1] H. Tahani and J. Keller "Information Fusion in Computer Vision Using the Fuzzy Integral," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 3, pp. 733-741, May 1990.

[2] H. Tahani and J. Keller "The Generalized Fuzzy Integral and Its Application to Object Recognition," under review, IEEE Transactions on Systems, Man, and Cybernetics.

[3] D. Dubois and H. Prade, "A Class of Fuzzy Measures Based on Triangular norms," Inter. J. Gen. Systems, 8, pp. 43-61, 1982.

[4] S. Weber, "$\perp$-Decomposable Measures and Integrals for Archimedean t-Conorms $\perp$," J. of Math. Analysis and Applications, 101, pp. 114-138, 1984.

[5] M. Sugeno, "Theory of Fuzzy Integrals and Its Applications," Doctoral dissertation, Tokyo Institute of Technology, 1974.

[6] H. Tahani, "Uncertainty Modeling Using Triangular Norms and Triangular Conorms", Proceeding, North American Fuzzy Information Processing Society-91, Columbia, MO, pp. 18-21, May 1991.

[7] S. Wierzchon, "On Fuzzy Measure and Fuzzy Integral," Fuzzy Information and Decision Process, M. Gupta and E. Sanches(eds.) North-Holland Publishing Company, pp. 79-86, 1982.

[8] J. Keller, H. Qiu, and H. Tahani, "Fuzzy Integral and Image Segmentation," Proc. North American Fuzzy Information Processing Society, New Orleans, pp. 334-338, June 1986.

[9] H. Qiu and J. Keller, "Multispectral Image Segmentation Using Fuzzy Techniques," Proc. North American Fuzzy Information Processing Society, Purdue University, pp. 374-387, May 1987.

[10] H. Tahani and J. Keller, "Automated Calculation of Non-additive Measures for Object Recognition", SPIE's Conference on Intelligent Robots and Computer Vision IX: Algorithms and Technique, Boston, MA, November 1990.

[11] H. Tahani, "The Generalized Fuzzy Integral in Computer Vision", Doctoral dissertation, University of Missouri-Columbia, May 1991.

Appendix

The t-conorms used in this paper:

1. Drastic sum

$$S_{\mathcal{D}}(a,b) = \begin{cases} a, & b = 0, \\ b, & a = 0, \\ 1, & a,b > 0. \end{cases}$$

2. Bounded sum

$$S_{\mathcal{B}}(a,b) = \min(1,a+b)$$

3. Algebraic sum

$$S_{\Pi}(a,b) = a + b - ab$$

4. Logical sum

$$S_{\mathcal{M}}(a,b) = \max(a,b)$$

The t-norms used in this paper:

1. Drastic product

$$T_{\mathcal{D}}(a,b) = \begin{cases} a, & b = 0, \\ b, & a = 0, \\ 1, & a,b > 0. \end{cases}$$

2. Bounded product

$$T_{\mathcal{B}}(a,b) = \max(0,a+b-1)$$

3. Algebraic product

$$T_{\Pi}(a,b) = ab$$

4. Logical product

$$T_{\mathcal{M}}(a,b) = \min(a,b)$$

N93

29569

UNCLAS

ON THE EVALUATION OF FUZZY QUANTIFIED QUERIES IN A DATABASE MANAGEMENT SYSTEM

Patrick BOSC & Olivier PIVERT
IRISA/ENSSAT
BP 447
22305 Lannion Cédex
FRANCE

ABSTRACT

Many propositions to extend database management systems have been made in the last decade. Some of them aim at the support of a wider range of queries involving fuzzy predicates. Unfortunately, these queries are somewhat complex and the question of their efficiency is a subject under discussion. In this paper, we focus on a particular subset of queries, namely those using fuzzy quantified predicates. More precisely, we will consider the case where such predicates apply to individual elements as well as to sets of elements. Thanks to some interesting properties of α-cuts of fuzzy sets, we are able to show that the evaluation of these queries can be significantly improved with respect to a naïve strategy based on exhaustive scans of sets or files.

I. INTRODUCTION

The database management systems currently available are based on the relational model and they suffer several limitations regarding user or application needs. In particular, it is assumed that data are precisely known (or fully unknown) and queries are based on crisp conditions. The notion of imprecision can be introduced in such systems at two levels : for representing imprecise or uncertain data and to allow flexible queries. In this paper, we will only consider the second aspect, that is to say that the data are assumed to take their values in ordinary universes, whereas queries may contain imprecise conditions. In this way, regular data bases are taken into account and the users are provided with answers consisting of an ordered list of elements (tuples) according to their adequation.

Various kinds of compound fuzzy predicates have been proposed in recent years [4, 8]. Base predicates described as fuzzy sets (i.e by means of characteristic functions) can be altered by linguistic modifiers and arranged together using connectors or aggregates in order to reach the appropriate semantics. Depending on the context, a predicate may apply to individual tuples or to sets of tuples; in both cases, a problem of performance is posed if the number of tuples is large and an exhaustive scan is performed. In a previous paper [3], we concentrated on the evaluation of compound fuzzy predicates applying to individual tuples. In particular, we showed that the computation of an alpha-cut of a fuzzy set could be performed in two steps : efficient selection of a superset of the alpha-cut by means of a boolean condition followed by the computation of the alpha-cut itself from this superset. In this paper, we deal with the evaluation of fuzzy quantified predicates which can concern either individual tuples or sets of tuples. Fuzzy quantifiers were first introduced by L.A. Zadeh [10] to generalize the existential (∃) and universal (∀) quantifiers. Recently, R. Yager suggested another approach to the definition of fuzzy quantifiers [7, 8, 9]. Our aim is to point out some efficient strategies for the evaluation of fuzzy quantified predicates, since efficiency is a key point in DBMS's [5].

In section 2, fuzzy quantified predicates are introduced along with their two possible interpretations. Their use in the framework of an extended relational language is also illustrated. In section 3, we point out some interesting properties of the OWA aggregation operator which will be useful in improving the evaluation. The evaluation of fuzzy quantified predicates applied to individual tuples and sets of tuples is discussed in sections 4 and 5 respectively. Starting from a naïve strategy based on an exhaustive scan of the concerned elements, we point out some properties intended for limiting the data to be accessed (and consequently the I/O volume). To conclude, we summarize the main results and draw some directions for future work.

II. FUZZY QUANTIFIED PREDICATES AND THEIR INTERPRETATION

2.1. Tuple and set oriented predicates

In the usual relational framework, we can distinguish predicates P applying to individual elements (x's) of a set X :

$$P : x \in X \rightarrow [0,1]$$

and predicates whose argument is a whole set X of elements :

$$P : X \rightarrow [0,1].$$

Typical examples of these two categories in an SQL language are :

"find the employees earning more than $4000" expressed :

```
select * from EMPLOYEE where salary > 4000
```

and "find the departments where the average of the salaries is over $4000" expressed :

```
select dep from R group by dep having avg(salary) > 4000.
```

2.2. Fuzzy quantifiers according to Zadeh

In the context of an extended relational language supporting imprecise querying such as SQLf [1, 2], it seems natural to introduce fuzzy quantifiers inside queries. The initial quantified propositions made by L.A. Zadeh were basically applying to sets of tuples [10] as shown in the following query example :

"find the best 10 departments where at least three employees are middle-aged" expressed :

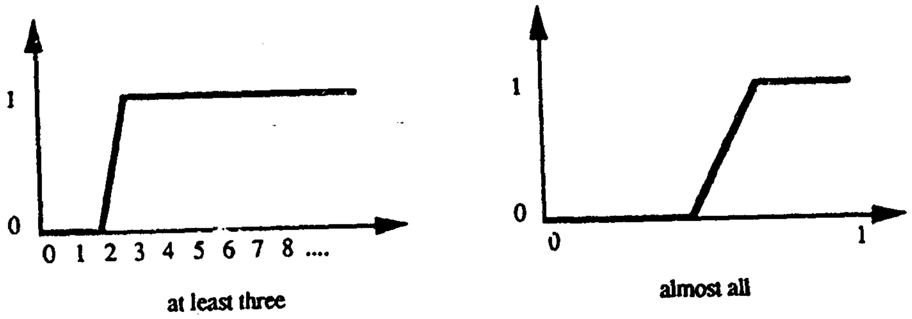
```
select 10 dep from EMPLOYEE                                        (A).
group by dep having at least three are middle-aged
```

Afterwards, J. Kacprzyk suggested an adaptation for individual tuples [6] inside queries such as :

"find the best 10 employees matching almost all of the predicates {middle-aged, really well-paid, ... }" expressed :

```
select 10 * from EMPLOYEE                                          (B).
where almost-all among {middle-aged, really well-paid, ...}
```

In both cases, the quantifier is seen as a fuzzy set defined on the cardinality of a fuzzy set. In example A, the quantifier is absolute (at least three) and the associated fuzzy set maps R into the unit interval [0,1]. If AQ stands for an absolute quantifier, the expression "AQ X are D" is interpreted according to the formula : $\mu_{AQ}(\Sigma\mu_D(x_i))$, where $\Sigma\mu_D(x_i)$ denotes the absolute cardinality of the fuzzy set associated to "X are D". In example B, we have a relative quantifier which is represented by an application from [0,1] to [0,1]. If RQ represents a relative quantifier, the expression "x matches RQ among $\{P_1, \ldots, P_n\}$" is defined as : $\mu_{RQ}((\Sigma\mu_{P_i}(x))/n)$. Possible shapes for the quantifiers used in these examples are given in figure 1.



at least three

almost all

Figure 1. Examples of the representation of two quantifiers.

2.3. Fuzzy quantifiers according to Yager

R. Yager recently suggested representing monotonous quantifiers by means of OWA aggregations [9]. First of all, let us recall the definition of an OWA :

$$\mathrm{OWA}(w_1, \ldots, w_n, x_1, \ldots, x_n) = \sum_{i=1}^{n} (w_i * x_{k_i}) \qquad (1)$$

where x_{k_i} is the i^{th} largest value among the x_i's.

Example. Let us consider the case where W = (.1, .2, .3, .4) and X = {.4, .9, .6, .1}. We will compute (.1 * .9) + (.2 * .6) + (.3 * .4) + (.4 * .1) and we get the value : .37.

In this context, a fuzzy quantifier is represented by the weights put into the operator, each of which expressing the contribution of the i^{th} largest element according to the figure drawn below :

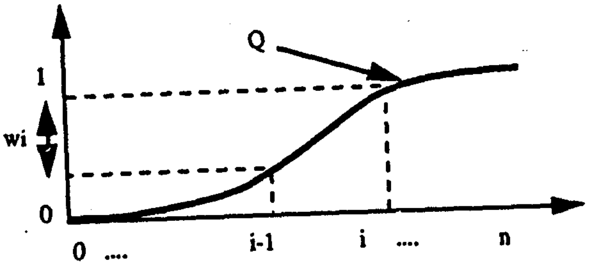


Figure 2. Weights design for a fuzzy quantifier.

2.4. Comparison of the two approaches

We will make the comparison in the context of an example. Let us consider the query : "find the sets where at least three elements are C" along with the two following fuzzy sets : {a/.9, b/.8, c/1, d/.1, e/.2, f/0} and {a'/.4, b'/.7, c'/.6, d'/.5, e'/.8} where the degree indicates the extent to which an element satisfies C. According to Zadeh's definition, the two sets have the same cardinality (3) and they will be considered equivalent whatever the characteristic function chosen for the quantifier. On the contrary, if we take an OWA interpretation with weights $w_1 = 1/3$, $w_2 = 1/3$, $w_3 = 1/3$, $w_i = 0\ \forall i > 3$, the degree for the first set is : .9 and that of the other is : .7. We prefer this second result since we believe that it wi!l better meet database users' intuition. The limitation is in the restriction of monotonous quantifiers, but the semantics of such an operator seems to be more convenient for database queries and this approach will be considered in the rest of the paper.

2.5. Queries under consideration

More precisely, we will concentrate on the evaluation of three types of queries :

- tuple-oriented (or horizontal) fuzzy quantified predicates :
  select ... from R where Q among {P_1, ... , P_n}

- type 1 set-oriented (or vertical-1) fuzzy quantified predicates :
  select ... from R group by att having Q are D

- type 2 set-oriented (or vertical-2) fuzzy quantified predicates :
  select ... from R group by att having Q (C are D)

In the first case, the weights are statically defined by the user (or an administrator) and the calculus is to be carried out according to formula (1). In the last two cases, the quantifier is assumed to be represented by a fuzzy set Q and the weights are calculated dynamically (depending on the set the quantifier applies to) and used in the following manner [9] :

- type 1 : i) $w_i = Q(i/n) - Q((i-1)/n)$;
  ii) compute $\Sigma w_i * \mu_D(x_{k_i})$ where x_{k_i} denotes the i^{th} largest value among the $\mu_D(x_i)$'s;

- type 2 : i) compute $y_i = \mu_C(x_i)$ for each x_i in X; call z_i the i^{th} smallest among the y_i's;

ii) $w_i = Q(\sum_{j=1}^{i} z_j / \sum_{i=1}^{n} z_i) - Q(\sum_{j=1}^{i-1} z_j / \sum_{i=1}^{n} z_i)$;

iii) compute $E_i = \max(\mu_D(x_i), \mu_{\overline{C}}(x_i))$ C are D is seen as the implication $C \Rightarrow D$;

iv) compute $\sum_x w_i * E_{k_i}$ where E_{k_i} represents the ith largest value among the E_i's.

In both cases, n is the cardinality of the set of tuples concerned with the quantification. For the sake of clarity and without loss of generality, we will assume that predicates involved in a quantification ($P_1, \dots, P_n$, C and D) are atomic fuzzy predicates.

Example. Let us consider the proposition "most of the middle-aged employees are well-paid" and let us assume a set of five employees e1 to e5 such that middle-aged = {.6/e1, .3/e2, 1/e3, 0/e4, .1/e5} and well-paid = {.8/e1, .4/e2, .9/e3, 1/e4, 1/e5}. Moreover, let "most of" be represented by the function : $x \rightarrow x^2$. The weights will be : $w_1 = 0$, $w_2 = .0025$, $w_3 = .0375$, $w_4 = .21$, $w_5 = .75$. The value of "middle-aged ⇒ well-paid" for each employee is : {.8/e1, .7/e2, .9/e3, 1/e4, 1/e5} and the final value of the proposition is given by : $(1 * 0) + (1 * .0025) + (.9 * .0375) + (.8 * .21) + (.7 * .75) = .72925$.

III. SOME PROPERTIES OF THE OWA OPERATOR

The idea which will be developed in the rest of the paper is situated in the scope of the evaluation of such predicates for a given threshold λ (satisfaction degree). The reason for that is the fact that in general, the user is only interested in a small subset of tuples and more precisely the best ones. In any case, we are only interested in those tuples whose degree is over 0, consequently, 0 is a lower bound for λ. In the following sections we will examine how to evaluate horizontal and vertical fuzzy quantified predicates where the quantifier is expressed by means of an OWA operator and we will also take advantage of properties of such an operator.

The OWA is a mean operator and so it has interesting properties :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_n) \le \max(x_1, \dots, x_n) \quad (2)$$

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, \dots, x_n) \le \mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, 1 \dots, 1) \quad (3)$$

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, \dots, x_n) \le \mathrm{OWA}(w_1, \dots, w_n, 1, \dots, 1, x_i, 1, \dots, 1) \quad (4)$$

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, \dots, x_n) \ge \mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, 0 \dots 0) \quad (5)$$

Properties (4) and (5) arise from : i) the monotonicity of any mean operator, ii) the fact that x belongs to [0,1]. From these basic properties, one can derive conditions bearing on the x_i's which are necessary for the satisfaction of the condition :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_n) \ge \lambda.$$

From (2), one can assert :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_n) \ge \lambda \Rightarrow \max(x_1, \dots, x_n) \ge \lambda \Leftrightarrow \exists i, x_i \ge \lambda \quad (6).$$

From (4), one can derive :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_n) \ge \lambda \Rightarrow \mathrm{OWA}(w_1, \dots, w_n, 1, \dots, 1, x_i, 1, \dots, 1) \ge \lambda$$

$$\Leftrightarrow (\sum_{i=1}^{n-1} w_i * 1) + w_n * x_i \ge \lambda$$

$$\Leftrightarrow (1 - w_n) + w_n * x_i \ge \lambda$$

and finally, we get :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_n) \ge \lambda \Rightarrow \forall i, x_i \ge \frac{\lambda + w_n - 1}{w_n} \quad (7).$$

This last formula is valid only if w_n is strictly positive, otherwise no implication can be found. Moreover, it is only profitable if $(\lambda + w_n) > 1$ (otherwise, we have a condition which is trivially satisfied).

From (3) and (5), we have :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, 1, \dots, 1) < \lambda \Rightarrow \mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_n) < \lambda \qquad (8)$$

and :

$$\mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, 0, \dots, 0) \geq \lambda \Rightarrow \mathrm{OWA}(w_1, \dots, w_n, x_1, \dots, x_i, \dots, x_n) \geq \lambda \qquad (9).$$

These last two conditions will be used for partial evaluations of an OWA aggregation as we will see in section 5.

IV. EVALUATION OF TUPLE-ORIENTED FUZZY QUANTIFIERS

4.1. Initial strategy

Let us consider the evaluation of quantified conditions applying to individual tuples of the form (in SQLf) :

select ... **from** R **where** Q **among** $\{P_1, \dots, P_n\}$.

The principle is to compute the sum $\Sigma_i (w_i * P_i(x))$ and a naïve algorithm could be :

```
for each r in R do
        for i from 1 to n do $V[i] = \mu_{P_i}(r)$ enddo;
        order the vector V giving V';
        GV = 0;
        for i from 1 to n do $GV = GV + V'[i] * W[i]$ enddo
        if $GV \geq \lambda$ then write(x/GV) endif
enddo
```

4.2. Improvements

This algorithm is based on an exhaustive scan of relation R (in practice the entire file containing relation R should be read) and it can be improved in two ways : the number of elements to be accessed and the calculations to be performed on each element. It could be of interest to replace the outer loop by : "**for each** r **in** R' **do**", as far as R' is a small subset of R which can be easily obtained from R (and efficiently, for instance without requiring the exhaustive scan of R) and it is sure that no possibly satisfying tuple has been discarded. At this point, we can come back to properties (2) and (3) and profit from the derived formulae (6) and (7) to proceed in two steps : creation of a subset R' of R by means of a **usual boolean condition**, then the application of the previous algorithm on R'. One expected interest of the first step is the fact that a regular DBMS is able to work efficiently as far as **indexes** or **access paths** are available. Formula (6) becomes :

$$P_1(r) \geq \lambda \text{ or } \dots \text{ or } P_n(r) \geq \lambda$$

and if we assume that each fuzzy predicate P_i is represented by a trapezium on an attribute A_i, we finally get a condition :

$$(r.A_1 \in [i_1, s_1]) \text{ or } \dots \text{ or } (r.A_n \in [i_n, s_n]) \qquad (8)$$

where i_j and s_j are the inferior and superior values associated to the λ-level cut of P_i. In a way similar, formula (7) (if $(\lambda + w_n) > 1$ and $w_n > 0$) leads to the condition :

$$(r.A_1 \in [i'_1, s'_1]) \text{ and } \dots \text{ and } (r.A_n \in [i'_n, s'_n]) \qquad (9)$$

where i'_j and s'_j are the inferior and superior values associated to the $((\lambda + w_n - 1)/w_n)$-level cut of P_i.

Since these two conditions are necessary, we can combine them together by means of a conjunction. However, it should be noted that condition (8) is a disjunction and if used alone, it will generally require the entire scan of the relation R to be executed and in this case, data access is not improved at all. On the other hand, condition (9) is conjunctive and if one of the attributes A_i is indexed, the whole condition can be processed with a limited number of data access.

4.3. An example

Consider the membership functions drawn in figure 3 in the scope of the query :

select * from EMPLOYEE **where**
most {middle-aged, high-salary, low-commission, medium-sales, around(nb-children,2)}.

According to the representation of "most" by the function : x → x^2, the weight vector associated to the OWA is : w_1 = .04, w_2 = .12, w_3 = .2, w_4 = . 28, w_5 = .36. If we assume that we want the tuples satisfying this quantified predicate at a level greater or equal to .82, we have to consider the operation :
OWA(.04, .12, .2, .28, .36, middle-aged(e), high-salary(e), low-commission(e), medium-sales(e), around(nb-children(e),2)) ≥ .82 for each employee e.

In this case, expression (8) becomes :

e.age ∈ [39,51] or e.salary ≥ 46000 or e.commission ≤ 8500 (10)
or e.sales ∈ [1.3,2.7] or e.nb-children = 2.

Since (λ + w_5) = 1.18 is greater than 0, formula (7) is applicable, (λ + w_5 - 1)/w_5 = .18/.36 = .5 and expression (9) yields:

e.age ∈ [37,53] **and** e.salary ≥ 41000 **and** e.commission ≤ 9800 (11)
and e.sales ∈ [1.2,2.8] **and** e.nb-children ∈ [1,3].

From a practical point of view, expressions (10) and (11) can be used in two different contexts or architectures. The first one is a pre-processing based on an explicit query submitted to a regular DBMS such as :

select * from EMPLOYEE e **where** e.age ∈ [37,53] **and** e.salary ≥ 41000 **and**
e.commission ≤ 9800 **and** e.sales ∈ [1.2,2.8] **and** e.nb-children ∈ [1,3] **and**
(age ∈ [39,51] or salary ≥ 46000 or commission ≤ 8500 or sales ∈ [1.3,2.7] or nb-children = 2).

The second one would consist of using these conditions at the internal level of query processing inside an extended DBMS able to evaluate fuzzy queries directly.

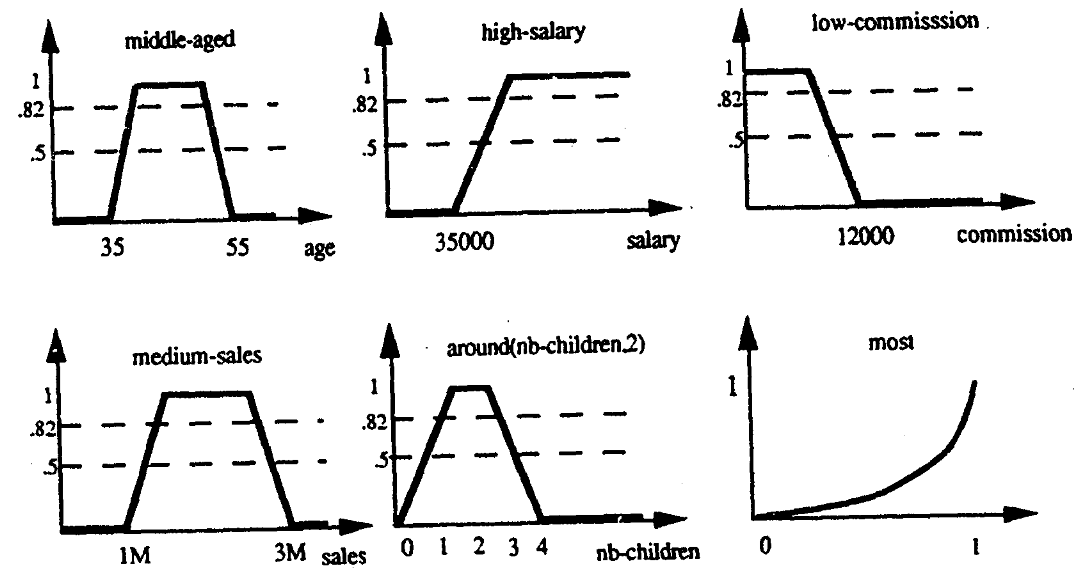


Figure 3. Membership functions used in the example.

V. EVALUATION OF SET-ORIENTED FUZZY QUANTIFIERS

5.1. Partitioning of relations and initial strategy

In section 3, the principle for the evaluation of fuzzy quantified predicates applying to one (or several) set(s) of tuples was given. Hereafter, we will consider queries of the form "**select** att **from** R **group by** att

having Q are D" (including a so-called vertical-1 fuzzy quantified predicate) where Q is a proportional quantifier which will be represented by weights appearing in an OWA aggregation. The "group by" clause in SQL is called partitioning and its meaning is explain-d in figure 4.

We now outline a naïve algorithm performing the calculus, assuming that R_j denotes the subset (or partition) of R where att = v_j :

```
for each $r_i$ in $R_j$ do $V[i] = \mu_D(r_i)$ enddo;
order the vector V giving V';
compute the vector W;
comment this calculus only needs the knowledge of n the cardinality of $R_j$ endcomment;
GV = 0;
for i from 1 to n do $GV = GV + V'[i] * W[i]$ enddo;
if $GV \geq \lambda$ then write($v_j$/GV) endif;
```

This algorithm looks similar to the previous one, but it is in fact very different. On the one hand the values used in the aggregation come from the tuples of a partition (issued from the grouping mechanism) and not from a single tuple, on the other hand the weight vector depends on n, the number of tuples of the considered subset of relation R. Therefore, it is clear that if n cannot be known without scanning a whole partition, no significant improvement regarding data access can be attained. So, we will assume that : i) the number of elements of each partition is known (this is possible especially if the relation is indexed on the attribute att or if the partitions are built using a sort which maintains the size of blocks with identical sort key value), ii) each partition can be accessed separately but step by step as tuples are required.

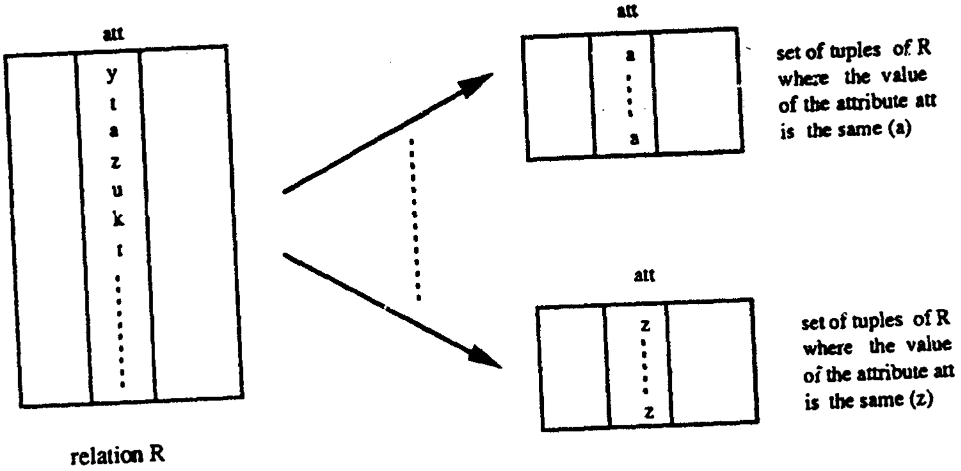

att
y
t
a
z
u
k
t
relation R
att
a
a
set of tuples of R where the value of the attribute att is the same (a)
att
z
z
set of tuples of R where the value of the attribute att is the same (z)

Figure 4. The "group by" mechanism in SQL.

5.2. Improved algorithm

Our aim is to reduce the access to tuples and by doing this, to indicate those conditions deciding whether the calculus should continue or is able to be stopped. More precisely, the calculus (mainly the loop which encompasses data access) can stop in two circumstances : i) when the partition cannot reach the desired level (λ), ii) when it is certain that the partition will reach the desired level (λ) and the precise value of the membership degree is not required. This reasoning is very similar to what is done in the design of "try and error" or "branch and bound" algorithms where some heuristics is searched in order to limit the number of candidates to be examined.

Since n is known, the weight vector W can be calculated and in particular its last value w_n . Thus, if the sum $(\lambda + w_n)$ is greater than 1, we can apply the condition (7) to any tuple r_i of a partition and insert the following instruction :

if $\mu_D(r_i) < (\lambda + w_n - 1) / w_n$ **then exit endif**

From a practical point of view, partitions with a large number of elements will lead to a low value for w_n and consequently this condition will not work frequently (except if λ is very close to 1).

Now, let us assume that we have already accessed k tuples of a partition (tuples r_1 to r_k), and the values $V[i] = \mu_D(r_i)$ for $i \in [1,k]$ are known. If we assume that the (n - k) missing values are 1 and the result of the OWA aggregation remains under λ, according to formula (8) we can be certain that this partition will never reach the desired level λ. We have to determine the aggregation :

$$OWA(w_1, \dots, w_n, v_1, \dots, v_k, 1, \dots, 1) = (\sum_{i=1}^{n-k} w_i) + \sum_{i=1}^{k} w_{n-k+i} * v_{j_{n-k+i}}$$

This computation requires only that the values V[1] to V[k] are sorted. In addition, the expression has not to be calculated from scratch from step k to step (k + 1) since for the first part, it is enough to subtract w_{n-k}. Thus, once again, we can specify a condition likely to stop the outer loop :

```
insert $\mu_D(r_k)$ into V[1:k] ;
compute $A = (\sum_{i=1}^{n-k} W[i]) + \sum_{i=1}^{k} W[n-k+i] * V[i]$;
if A < λ then exit endif
```

When k = n (last tuple of the partition), the value of A equals precisely the value GV which is the degree tied to the partition.

Finally, if the value of the membership degree of the partition is not necessary (for instance, the query looks only for the best p partitions but does not aim at ordering them), we can take advantage of formula (9). In fact this formula states that if we have already accessed k tuples of a partition (tuples r_1 to r_k) and we assume that the (n - k) missing values equal 0 and the result of the OWA aggregation already exceeds λ, then we can be sure that this partition will reach this desired level λ. We have to determine :

$$OWA(w_1, \dots, w_n, v_1, \dots, v_k, 0, \dots, 0) = \sum_{i=1}^{k} w_i * v_{k_i}$$

Here again, it is just necessary that the values V[1] to V[k] are sorted and we can specify a condition likely to stop the outer loop :

```
insert $\mu_D(r_k)$ into V[1:k] ;
compute $B = \sum_{i=1}^{k} W[i] * V[i]$;
if B ≥ λ then write($v_j$); exit endif
```

Again, when k = n, the value of B equals that of A and is GV the membership degree of the current partition.

We can now give the final algorithm, when n the number of tuples of any partition, is known in advance :

```
A = 0; compute the vector W; comment W[i] = Q(i / n) - Q((i - 1) / n) endcomment;
for each $r_k$ in $R_j$ do
        if $\mu_D(r_k) < (\lambda + w_n - 1) / w_n$ then exit endif;
        insert $\mu_D(r_k)$ into V[1:k] ; comment in decreasing order endcomment;
        $A = (\sum_{i=1}^{n-k} W[i]) + \sum_{i=1}^{k} W[n-k+i] * V[i]$;
        if A < λ then exit endif;
        $B = \sum_{i=1}^{k} W[i] * V[i]$ ;          comment these two instructions are present only if we
        if B ≥ λ then write($v_j$);      are not interested in the membership degree endcomment;
                      exit
        endif
enddo;
if A ≥ λ then write($v_j$/GV) endif;         comment this instruction appears if the previous do not
                                             endcomment;
```

Finally, we have to deal with the last case, namely the so-called vertical-2 fuzzy quantified predicates (Q r's C are D). If we look at the definition given in section 2, we can see that the weights are dependant on the value of the $\mu_C(x_i)$'s and it does not seem realistic to perform this calculus without the entire scan of the underlying relation R. Consequently, the canonic algorithm derived from the definition can be applied and it will require the exhaustive scan of all the partitions created by the "group by".

5.3. An example

Let us consider the query "find the best 5 departments where most of the employees are well-paid" which is expressed in SQLf as :

select 5 dep from EMPLOYEE group by dep having most are we!l-paid.

We examine a department (partition) containing five employees e1 to e5 with the following characteristics :

#emp	#dep	salary	...
e2	d	38000	.
e4	d	55000	.
e1	d	46000	.
e5	d	32000	.
e3	d	48000	.

with "most" represented by the function : x → x^2, λ is set to .73 and "well-paid" is the membership function given in figure 5.

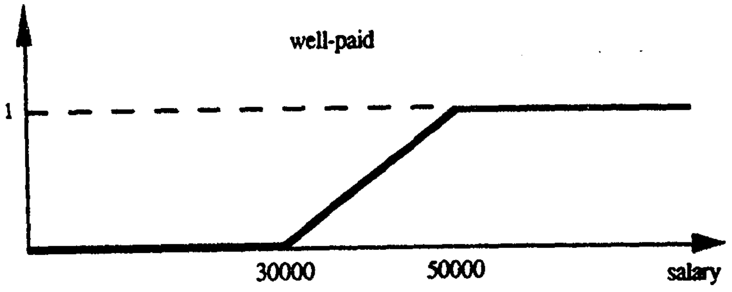


Figure 5. The membership function for the predicate "well-paid".

So, the fuzzy set well-paid is {.8/e1, .4/e2, .9/e3, 1/e4, .1/e5} and the weight vector W is : w_1 = .04, w_2 = .12, w_3 = .2, w_4 = . 28, w_5 = .36. If we perform the overall calculus for these data (naïve strategy requiring the access to the 5 tuples), we get : (.04 * 1) + (.12 * .9) + (.2 * .8) + (.28 * .4) + (.36 * .1) = .456; therefore, this partition does not match our requirement (.73). Now, let us apply our improved algorithm assuming that the tuples are accessed according to the order depicted above. Since ($\lambda + w_5$) is over 1 (($\lambda + w_5 - 1) / w_5$ = .25), the first condition of the algorithm is interesting (not trivially satisfied).

Access employee e2 : $\mu_{well\text{-}paid}(e2)$ = .4 > .25; A = .784 > .73; B = .016 < .73 ⇒ the loop goes on
Access employee e4 : $\mu_{well\text{-}paid}(e4)$ = 1 > .25; A = .784 > .73; B = .088 < .73 ⇒ the loop goes on
Access employee e1: $\mu_{well\text{-}paid}(e1)$ = .8 > .25; A = .728 > .73 is false ⇒ **the loop stops here.**

In this case, we save 2 accesses and if e5 were the first tuple of the considered partition, the loop would have stopped immediately, since μwell-paid(e5) = .1 is under .25, and 4 data accesses would have been saved.

VI. CONCLUSION

In this paper, we have dealt with database management systems where conventional data are stored and support imprecise queries. More precisely, we have concentrated on fuzzy queries involving quantifiers. We have distinguished two main classes of such queries : 1) those where the quantified condition applies to each element of a set (x matches Q among {P_1, ... , P_n}), and 2) those where the quantified condition concerns a whole set of elements : a) Q x's are D, or b) Q x's C are D. Two major frameworks for the interpretation of quantifiers exist and we have chosen Yager's approach which is based on the OWA aggregation operator, since we believe that generally it fits more the appropriate semantics of a database query.

Our objective was to design some strategies for the evaluation of these queries, when a threshold for the satisfaction degree is given by the user. Starting from a naïve strategy based on the exhaustive scan of the considered set, we have pointed out some properties of the OWA operator allowing for some improvements especially regarding data access. For type 1 queries, a boolean query selecting a subset of the elements likely to be satisfactory can be evaluated and if appropriate indexes are available, it is then possible to save data accesses. For type 2a queries, where the number of elements of the concerned set is known, we have shown that conditions could be applied to each element of the set to decide whether or not the calculus had to be continued. This approach is very similar to the introduction of heuristics in "try and error" or "branch and bound" algorithms. The basis of the improvements is to reduce the number of data accesses. Consequently, we replace algorithms in θ(n) by algorithms in O(n) and the class of complexity of the final algorithms has not changed. Finally, it does not seem that type 2b queries can be significantly improved.

One interesting result of this work is to show that a fuzzy query is not necessarily inefficient even for queries involving quantifiers. Moreover, the notion of heuristics used for type 2a queries is likely to work for other kinds of set-oriented queries such as those where a fuzzy predicate applies to a monotonic aggregate (sum or average on positive values for instance). In the near future, we will perform some simulations in order to get an idea about the gain provided by our improvements.

REFERENCES

[1] P. Bosc, M. Galibourg, G. Hamon, "Fuzzy querying with SQL : extensions and implementation aspects", *Fuzzy Sets and Systems*, **28**, 333-349, 1988.

[2] P. Bosc, O. Pivert, "About equivalences in SQL^f, a relational language supporting imprecise querying", *Proc. International Fuzzy Engineering Symposium*, Yokohama (Japan), 309-320, 1991.

[3] P. Bosc, O. Pivert, "Some properties of alpha-cuts of fuzzy predicates", *Proc 11th European Meeting on Cybernetics and Systems Research*, Vienna (Austria), 1992.

[4] D. Dubois, H. Prade, "A review of fuzzy set aggregation connectives", *Information Sciences*, **36**, 85-121, 1985.

[5] M. Jarke, J. Koch, "Query optimization in database systems", *Computing Surveys*, **16**, 111-152, 1984.

[6] J. Kacprzyk, A. Ziolkowski, "Database queries with fuzzy linguistic quantifiers", *IEEE Transactions on Systems, Man and Cybernetics*, **16**, 474-478, 1986.

[7] R.R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking", *IEEE Transactions on Systems, Man and Cybernetics*,**18**, 183-190, 1988.

[8] R.R.Yager, "Connectives and quantifiers in fuzzy sets", *Fuzzy Sets and Systems*, **40**, 39-75, 1991.

[9] R.R. Yager, "Fuzzy quotient operators for fuzzy relational databases", *Proc. International Fuzzy Engineering Symposium*, Yokohama (Japan), 289-296, 1991.

[10] L.A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages", *Computer Mathematics with Applications*, **9**, 149-183, 1983.

N93
29570
UNCLAS

A Fuzzy Case Based Reasoning Tool for Model Based Approach to Rocket Engine Health Monitoring

Srinivas Krovvidy, Adam Nolan, Yong Lin Hu and William G. Wee
Artificial Intelligence and Computer Vision Laboratory
University of Cincinnati
Cincinnati, Ohio 45221-0030

Tel: (513) 556 4778
Fax: (513) 556 7326
Email: wwee@uceng.uc.edu

I. INTRODUCTION

One of the main requirements of a rocket engine Health Monitoring (HM) system is its ability to recognize potential failures of all kinds such that catastrophic failures can be avoided through cutoff and other less catastrophic failures can be avoided through repair works. The HM system must have the ability to learn new situations and be able to recognize potential failures. The behavior of key SSME performance parameters vary significantly depending on engine power level and changing interface conditions (Nemeth et al., 1990, Millis 1991). Parameters included in this list are turbine discharge temperatures, other turbopump inlet and discharge temperatures and pressures, turbopump speeds, propellant flow rates, and valve positions. Therefore, a model based approach is well suited to identify dynamic, nominal operating values. In real HM operation, we are always confronted with uncertain data, data where event of physical failures occurs. A fuzzy set approach (Kosko, 1992) to describe this data is most logical.

In the recent years, researchers are investigating a new paradigm for problem solving and learning, by using specific solutions to specific situations (Riesbeck and Schank, 1989). The basic idea is to make use of the old solutions while solving a new problem, and such an approach is known as Case Based Reasoning (CBR)(Krovvidy & Wee, 1992, Riesbeck & Schank, 1989). A model based approach is found to be one of the useful approaches for designing planning systems (Birnbaum et al., 1991). Currently rocket engine protection consists of redline systems that issue an engine cutoff if measured value exceeds a pre-determined operation limit for any of several parameters (Millis, 1991). More recently efforts

5

are being made to develop an advanced framework for a failure detection system with the addition of model ba:ed algorithms (Hawman et al., 1991).

In this paper, we develop a fuzzy case based reasoner that can help building such a model from old cases and any existing domain knowledge. A detailed system description is presented in this paper.

II. PROBLEM ST \TEMENT AND SUGGESTED APPROACH

In this system we develop a fuzzy case based reasoner that can build a case representation for several past anomalies detected, and develop case retrieval methods that can be used to index a relevant case when a new problem (case) is presented using fuzzy sets. The choice of fuzzy sets is justified by the uncertain data. The new problem can be solved using knowledge of the model along with the old cases. This system can then be used to generalize the knowledge from previous cases and use this generalization to refine the existing model definition. This in turn can help to detect failures using the model based algorithms.

III. SYSTEM DESCRIPTION

The proposed Fuzzy Case Based Reasoner (FCBR) is depicted as shown in the Figure 1.

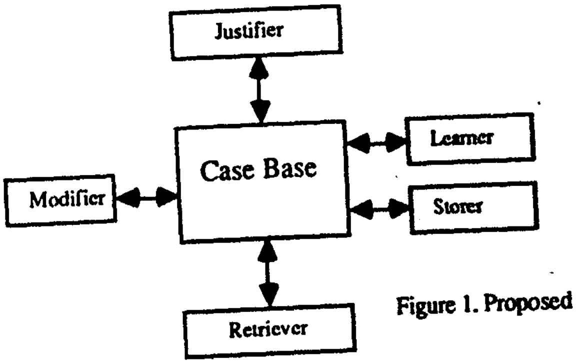


Figure 1. Proposed Fuzzy Case Based Reasoner

A case is defined as an n-vector (vector of n dimensions from n sensors) with m samples over a time window T seconds. This n by m matrix constitutes a case input to be either trained (learned) or decided (tested). This case definition allows a decision to be generated or made every T data width of m samples. It is therefore possible to generate a decision per sample interval $\Delta T = T/m$. But in general, to be practical, a multiple of ΔT say

$k\Delta T$ with $k << m$ can be used as the decision interval. We need to be sure that m samples over T seconds is enough to model both the engine start-up and shut-down, and main stage operation.

It is very difficult to give a definite relationship between data collected and the fault occurring at a given time and at a specific location because of many uncertainties. A monotonic fuzzy number is modelled using time and sensor location for each case. This modelling in discrete time space will be used in generating both training and testing cases. In training, a fuzzy number from [0-1] is generated and associated with the data of an n by m matrix. In the testing phase, we need to consider multiple decisions under different time scales. When a new problem is given, we use FCBR to find the closest case from the previous cases. We will predict the chances of failure at different future time periods and then propose a general decision scheme for the given case from those predictions.

IV COMPONENTS OF FCBR

1) **Retriever, Modifier, Justifier, Storer and Learner:** In diagnostic design problems, case retrieval should be done based on the qualitative description of the problem and the causal relations in the explanation of the design solution. The indexing mechanism must also allow cues to access cases at any level of the representation using fuzzy set theorems. The criteria that are used to evaluate whether a case is similar enough to the current design problem should use the salient features of the domain. In the SSME problem, we must use the sensors' data to evaluate the applicability of an old case for a new problem.

If the retrieved solution is not acceptable the Modifier tries to adapt and synthesize different parts of the design into a solution using fuzzy sets. The Modifier can help us to suggest the necessary changes to be made in the dynamic modelling of the failure. The Justifier justifies the suggested solution.

The Storer stores the case. When a solution for a given problem is obtained it can be stored in the case base for future retrieval. When a set of sensor data is diagnosed for failure prediction, that data within the defined window (n by m matrix) must be stored in the case base. The cases would be diagnosed based on some monotonic fuzzy number. This value would define the chances of failure for that particular data set. Therefore, the cases are stored using fuzzy set concepts. The Learner then develops generalized solution strategies from the stored fuzzy cases. This is particularly important because of the enormous amount of data generated by the sensors. Therefore, we will develop generalization methods to take several cases and represent them in some form of rules so that we can contain the size of the case base.

V PRELIMINARY EXPERIMENTAL RESULTS

Some preliminary experiments are performed using the data from several sensors. In particular, we selected 4 sensors and defined a fuzzy case based reasoning system.

1) Data sets selection

In general the cases are defined based on multiple sensors. Our current test is restricted to the problem of detecting faults in the HPFTP (High Pressure Fuel Turbopump). Four sensors are selected and listed in the table 1.

ID	PID NO.	LABEL
1	7	MCC Pressure
2	17	HPFT Discharge Temperature
3	77	MCC Hot Gas Injector Pressure A
4	78	MCC Coolant Discharge Temperature B

Table 1. SELECTED SENSOR FOR CASE STUDY

The data sets of test 902-457, 902-463, and 901-463 are used in our current study. (Hawman et al., 1990) The tests 902-457, and 902-463 are two nominal data sets with no shutdown. The test 901-436 was reported with having a problem of HPFTP coolant liner buckle. It was shutdown due to a HPFT discharge temperature redline at t = 611.035 seconds.

2) Case definition

The sampling rate is defined as 0.04 which means T/0.04 samples are generated in T seconds. A case is defined as the samples generated in T seconds. In particular this is represented by a T x 4 vector. All the cases obtained from test 902-457 and 902-463 are considered to be safe. The cases obtained from test 901-436 has varying levels of failure modes. In other words the cases collected well before the breakdown have a low possibility of failure while those cases closer to the breakdown have a high possibility of failure.

3. Normalization of the data

Since the value of sensors highly depends on the power level, a normalization procedure corresponding to the power level is applied. The MCC pressure (MCC_PC) is proportional to the power level. It is used to define the measurement of power level. The power level (LP(t)) is defined as a ratio of MCC_PC value with predefined standard MCC_PC value (MCC_PC_STAND), $LP(t) = \frac{MCC\_PC(t)}{MCC\_PC\_STAND}$. The corresponding sensors level $CSL(i,t) = \frac{\text{Sensor Values}(i,t)}{\text{Sensor\_stand}(i)}$ can be estimated by some polynomial functions of power level (PL(t)) as follows:

$$CSL(i,t) = C1*PL^3(t) + C2*PL^2(t) + C3*PL(t) + C4$$

The coefficients (C1, C2, C3, C4) are obtained based on the nominal test data 902-457 with linear regression technique. The Sensor_stand(i) is standard value of sensor i which is predefined based on data from the nominal test 902-457 corresponding to the predefined standard MCC_PC value. The normalized value of each sensor are computed as follows:

$$norm(i,t) = \frac{\text{Sensor\_Value}(i,t)}{\text{Sensor\_stand}(i) * CSL(t)}$$

For our defined case matrix $A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}$ $A_i = (a_{i1}, a_{i2}, a_{i3}, ..., a_{iN})$, a normalized average percentage error (APERR) is defined as the case index.

$$APERR(t) = \frac{1}{4}\sum_{i=1}^{4} |1.0 - enorm(i,t)|$$

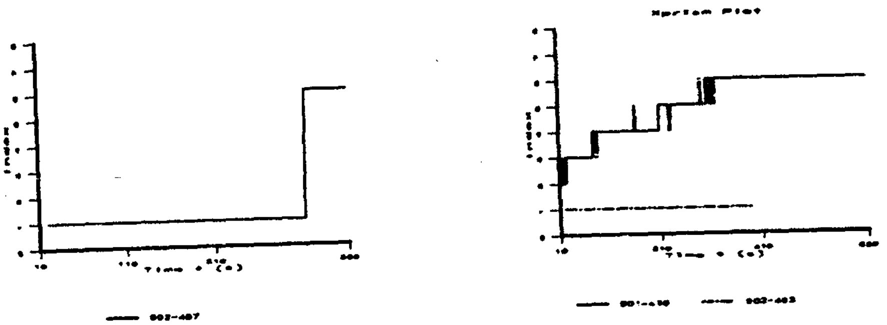
where

enorm(i,t) is average norm of sensor i within N data points

$$\mathrm{enorm}(i,t)=\frac{1}{N}\sum_{k=-\frac{N}{2}}^{k=\frac{N}{2}}\mathrm{norm}(i,t+kT)$$

N is number of samples within the window, T =0.04 is the sample rate

The cases are grouped such that they are classified into one of the categories { high risk, moderately risk, low risk and no risk}. The cases are stored in a case base. The retrieval from the case base is done using a hierarchical indexing. At the first level, we take the sample and compute its APERR. There we will be retrieving all those cases with a similar APERR. In the next step, we use a function defined on the first sensor data. The matching is continued until we identify the group to which the sample belongs. After obtaining the group, we can associate the possibility of breakdown with the new problem same as that of the identified group. The grouping of different cases is shown in the Figures 2 and 3. This has been identified as the primary index. With more sensors we expect to develop several such indexes and also more categories of cases. We also want to compare the results with other methods.



Figures 2. Grouping of cases in Breakdown and nonbreakdown data

The methods proposed were presented in the context of specific sensor data set analysis. The primary reason for this is to be able to compare recent performance (Hawman, et al., 1990) of regression analysis and linear predictors to that of the fuzzy case based reasoner. With adequate performance FCBR will be utilized as sensor models for several other parameters deemed relevant by the 1990 sensor study (Carter et al., 1990). This will enable the development of a fault detection system which would be less complex and more accurate than previously proposed methods.

The application of these methods are not isolated to SSME data. Success in this study implies wide ranging application to all engine monitoring systems.

V REFERENCES

Birnbaum, L. et al., *A Model-Based Approach to the Construction of Adaptive Case-Based Planning Systems*, Proceedings of Case-Based Reasoning Workshop, May 1991.

Carter, D., Denny, S.K. ,Rocket Engine Condition Monitoring System- Phase I, Technical Report CDRL 3012, Pratt&Whitney, United Technologies Corp. July 30, 1990.

Hawman, M.W., Galinaitis, W.S., Tulpule, S., Mattedi, A.K., Kamenetz, J., Framework for A Space Shuttle Main Engine Health Monitoring System, Final Report, NASA-CR-185224, United Technologies Research Center, March 1990.

Kosko, B., Neural Networks and Fuzzy Systems, Prentice Hall, 1992.

Krovvidy, S. and Wee, W.G. *Wastewater Treatment Systems from Case Based Reasoning,* Special issue on Case Based Learning of Machine Learning. (1992, in press)

Millis, M.G., *Technology Readiness Assessment of Advanced Space Engine Integrated Controls and Health Monitoring*, Proceedings of the Third Annual Health Monitoring Conference for Space Propulsion Systems, Cincinnati, Ohio, November 1991.

Nemeth, E., Maram, J. and Norman, Jr., A.M., *Health Management System for Rocket Engines*, Proceedings of the Second Annual Health Monitoring Conference for Space Propulsion Systems, Cincinnati, Ohio, November 1990.

Riesbeck, C.K., & Schank, R.C. *Inside Case-Based Reasoning* Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1989.

N93
29571
UNCLAS

N93-29571

A High Performance, Ad-Hoc, Fuzzy Query Processing System for Relational Databases

S15-82

161855

P-10

William H. Mansfield, Jr.
Bellcore, USA
whm@thumper.bellcore.com

Robert M. Fleischman**
BBN, USA
rmf@diamond.bbn.com

ABSTRACT

Database queries involving imprecise or fuzzy predicates are currently an evolving area of academic and industrial research [Buc87,Bosc88,Prad87,Tah77,Uma83,Zem85]. Such queries place severe stress on the indexing and I/O subsystems of conventional database environments since they involve the search of large numbers of records. The Datacycle™ architecture and research prototype is a database environment that uses filtering technology to perform an efficient, exhaustive search of an entire database. It has recently been modified to include fuzzy predicates in its query processing. The approach obviates the need for complex index structures, provides unlimited query throughput, permits the use of ad-hoc fuzzy membership functions and provides deterministic response time largely independent of query complexity and load. This paper describes the Datacycle prototype implementation of fuzzy queries and some recent performance results.

1. Introduction

In relational database systems [Codd70] databases contain tabular representations of information where rows represent database records (tuples) and columns represent fields (attributes) within the records. Relational algebra defines operations that can be carried out to specify particular query requests in which attribute values and Boolean logic are used to identify sets of records of interest. Structured Query Language (SQL) is a query language that defines the grammar and the user interface between an application and the database management system. In SQL, database data retrieval operations are defined in select statements of the form

Select attribute-list from relation where predicate

where the *attribute-list* identifies values to be returned to the user, *relation* identifies a particular table in the database, and the *predicate* identifies a search criteria consisting of Boolean expressions involving attribute names and values. One characteristic of these queries is that the user must be very familiar with the contents of the database, from both the perspective of structure, as well as the value range for particular attributes. Mechanisms to introduce meaningful imprecise terms into the predicate such as *young , old, high, and low* do not exist.

Fuzzy set theory [Zad65] has been proposed as one method for introducing imprecise queries into database systems. Efforts have been made to pre-process imprecise requests [Gala91],[East87] into a relational query language such as SQL or QUEL where a request for young employees might be translated into a range request for employees between the ages of 20 and 30.

Membership functions provide the method to translate an attribute value to a degree of membership in a fuzzy set, referred to as a possibility value. Figure 1 shows membership functions that map age values into the fuzzy sets YOUNG, MIDDLE AGE and OLD. Ages less than 15 are definitely members of the set YOUNG and have possibility values equal to 1.0. For ages between 15 and 25, the degree of membership

** Work was performed while the author was at Bellcore.

TM Datacycle is a trademark of Bellcore.

in the set YOUNG decreases from 1.0 to 0.0 indicating that ages closer to 15 are "younger" than ages closer to 25. For ages beyond 25, the function YOUNG yields possibility values equal to 0.0 indicating that over 20 ages are not members of the set YOUNG.

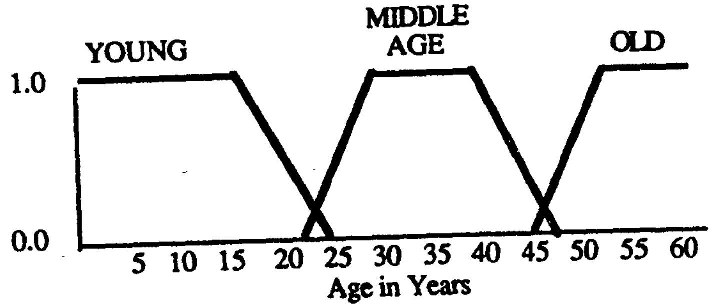


FIGURE 1
Membership functions for YOUNG, MIDDLE AGE and OLD

A common approach in implementing fuzzy query capability is to utilize specialized database index structures that associate records to fuzzy sets. The database index structures avoid the complexity of executing the membership function against every tuple in the database during query processing. This approach allows high-speed access for a predetermined number of fuzzy predicates. Arbitrary queries involving derived data negate the use of the these index structures and force the run-time execution of membership functions. This performance penalty is further impacted by the need to perform set intersection and union operations involving large sets.

The combined effects of these performance penalties make high performance fuzzy query systems difficult to implement. While the index approach improves response time performance, it reduces flexibility by limiting the user to a small number of pre-determined fuzzy sets. In contrast, to maximize query flexibility, a run-time execution of membership functions that can be specified within the query grammar is required. *Using the Datacycle prototype, we have recently implemented a fuzzy query capability that offers sub-second response time for large databases for a virtually unlimited number of concurrent users.* The approach permits the ad-hoc definition of membership functions in the query grammar, arbitrary use of numeric attributes in the database, and high performance.

The specification of the membership function at the grammar level coupled with the efficient run-time execution of the membership function are major contributions of this work. These two characteristics provide the primitives for adapting membership functions in either the database management system or within the application. Thus, an adaptive feedback loop can easily be implemented to support various learning algorithms to adjust the membership functions over time to the underlying database, or to accommodate shifting data. A second contribution of this work is the dynamic modification of membership functions to permit their use over very different data attributes.

Section 2 presents the Datacycle architecture and details of the prototype implementation in the context of the Datacycle project's original goals ("crisp" query processing). Section 3 describes fuzzy query extensions including the grammar and processing environment. Section 4 introduces a method for dynamically modifying the membership function to the underlying data during query execution. Section 5 provides performance measurements from benchmarking activities. Section 6 identifies some future work based on this approach, and Section 7 offers our conclusions.

2. The Datacycle™ Architecture and Research Prototype

The Datacycle architecture and research prototype [Her87,Bow91,Bow92] is a revolutionary approach to database processing motivated by the need to provide both high performance and flexible data access in single database system. This section describes the architecture and prototype in terms of processing standard relational (crisp) queries. Section 3 then builds on the architecture description and characteristics to describe our implementation of fuzzy queries.

In the Datacycle architecture, entire databases are broadcast over high bandwidth communications facilities to specialized filtering hardware [Lee91] that performs the complex data selection and aggregation operations necessary to satisfy application data requests. The original objectives of the project were to demonstrate extreme transaction throughput rates (tens of thousands of read transactions per second) in a system that also supported high update throughput. While offering flexible access to the data, a number of advantages of the filtering operation were largely unanticipated. The extension to include fuzzy query predicates is one example of the architecture's flexibility.

From the perspective of the various applications accessing it, the Datacycle experimental research prototype appears as a single database server with an SQL-based interface. Applications interact with database contents using ANSI SQL with the addition of some data manipulation primitives providing non-traditional functionality. Database contents are defined and managed solely in terms of the relational schema and the values of attributes of individual tuples; no distinction between "indexed" and "non-indexed" attributes exists. There are no indexes. *The entire database is content-addressable* meaning that records can be identified and retrieved based on the content of any attribute or combination of attributes. Thus, if a table contains 15 numeric attributes, any one, or several of the 15 can be used in a predicate with no performance penalty.

The internal Datacycle system model is depicted in Figure 2 and comprises of an arbitrarily large number of access managers acting on a single set of shared data items. The entire set of data items is maintained in a central subsystem call the storage pump. Data items are made available to the many access managers by repetitive broadcast of the entire contents of the database. On-the-fly search of the broadcast stream is the responsibility of custom VLSI datafilters within the access managers. Datafilters are essentially application specific microprocessors whose architecture and instruction set are optimized for synchronous, high speed search. The presence of the entire contents of the database on the broadcast channel provides the opportunity for direct selection of records based on the values of any attribute or combination of attributes, eliminating the need to store and maintain indices. The broadcast communication media allows access managers to be geographically distributed over wide areas. Database scaling is achieved using multiple pumps and their associated communication and filtering subsystems.

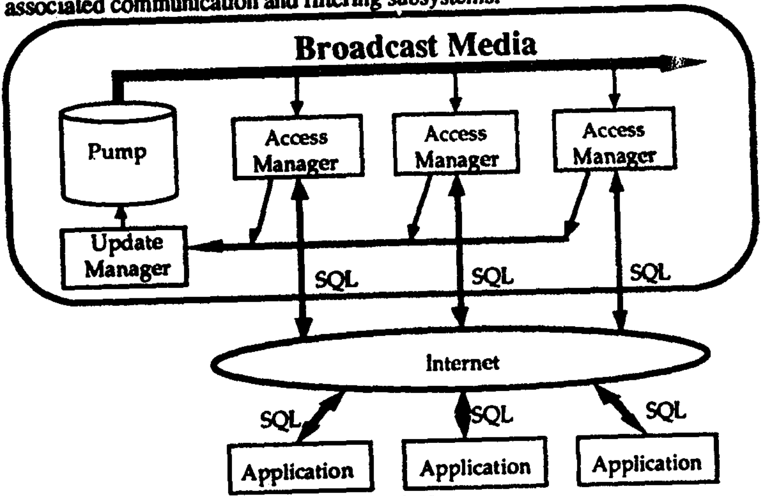


FIGURE 2.
The Datacycle Architecture

The custom VLSI datafilters have an instruction set that is optimized for Boolean comparison and arithmetic primitives. The filters are dual buffered devices allowing random access in the foreground buffer while a subsequent record is filling the background buffer. Thus, at the record level, filtering is not stream oriented. Thirty-two datafilter instructions can be executed while an individual record is present in the buffer. A single instruction is sufficient to complete a 4-Byte comparison and mark a record for selection, associate it with a specific query, and initiate output. Complex, multi-predicate selections or several independent selections can be performed simultaneously within the filter within the 32 instruction constraint. The datafilter instruction set includes arithmetic instructions that operate on integer data values. The ability to calculate numeric functions based on database contents provides the primitive operations necessary to perform membership functions on-the-fly while data records are present in the filter.

In the Datacycle experimental research prototype, the storage pump is implemented in a 32 - 128 MByte dual ported, banked RAM that allows the storage contents to be read sequentially for broadcast while portions of the database are available for update operations. The memory contents are broadcast over a 32 bit wide communication channel at 53 MBytes per second. A 16 MByte database will appear on the broadcast channel once every .3 seconds and the system will offer the user about 1 second response time for selects against the database. A 32 MByte database provides storage for 256K 128 byte tuples.

The content-addressability and full database scan, coupled with the flexibility of the filtering operation permit a variety of database selection operations that are particularly troublesome to conventional database system approaches. In an Operator Services (telephony) application setting, we have included longitude and latitude information for every customer in the database. Multi-dimensional range searches can be completed in a single broadcast cycle and spatial queries including CLOSEST (find the nearest object in the database) can be dealt with in two passes (one to identify the object and a subsequent pass to retrieve it). The CLOSEST function requires that a distance function be calculated on-the-fly within the filter. This calculation is representative of a larger set of selection operations that perform arithmetic transformations on one or more attributes. In conventional systems, these transformations often negate the advantages of traditional database index structures forcing a full database scan requiring extensive processing and causing extreme response time delays. In the Datacycle architecture, since a full database scan is always performed, variations in query complexity are often handled in constant response time

3. Fuzzy Queries within the Datacycle Experimental Prototype

We have recently completed an investigation of fuzzy query processing in the Datacycle architecture. Our work has centered on storing crisp database values and applying fuzzy query predicates during selection operations. Fuzzy requests define algorithmic membership functions that map the value of a database attribute to a degree to which it meets a fuzzy predicate. Fuzzy selection predicates include imprecise qualifiers such as *near*, *high*, *old*, *best*, *tall*, etc. Several of these membership functions can be combined using fuzzy logic to identify data objects that best meet a number of vague or imprecise selection specifications. For example, a fuzzy database request may ask for circuits with a *high* signal-to-noise ratio and a *low* maintenance history that terminate *near* a particular location. Such requests place a high degree of stress on the indexing and I/O operations in conventional database systems because they force the system to consider large numbers of tuples in a search to find an optimal, or some number of "best" matches. The Datacycle filtering primitives can perform the efficient evaluation of membership functions and the fuzzy logic necessary to combine them.

In the Datacycle prototype, we have chosen to utilize SQL extensions based on fuzzy queries consistent with previous fuzzy query grammars [Buc83,Kac89,Tah77, Zem85]. We have extended the grammar to allow the dynamic definition of membership functions from the application level.

Select * from R where att is qual   ***att = attribute name***
                                         qual = fuzzy term

The extended SQL query to select all the records for individuals in the fuzzy set YOUNG would be:

Select * from R where age is YOUNG

The fuzzy term can be defined as a trapezoid as depicted in the general case in Figure 3. The breakpoints {A,B,C,D} define the range (support) of the membership function. For cases other than the general case we have chosen to use a strict positioning of variables and the use of nulls for unspecified parameters (e.g. {,,C,D}). Other more natural alternatives [Zem85] for specifying these functions have been suggested. We chose this explicit notation to simplify parsing during query processing.

We have implemented a library of commonly used membership functions parameterized by these trapesoid breakpoints. During query parsing, breakpoints are substituted for recognized fuzzy qualifiers. To facilitate the ad-hoc definition of other fuzzy qualifiers, the breakpoints can be specified within the grammar. Thus using the breakpoints *{,,15,25}* instead of the fuzzy term *YOUNG*, the query to select entire record for all the young individuals named Smith would be

Select * where name=smith and age is { ,,15,25}

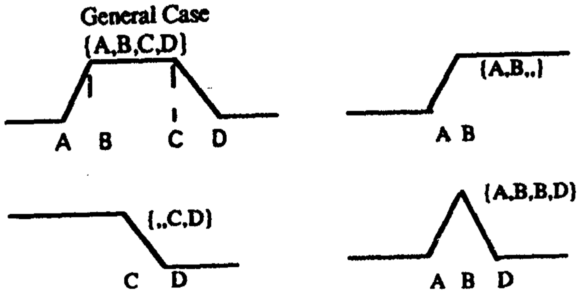
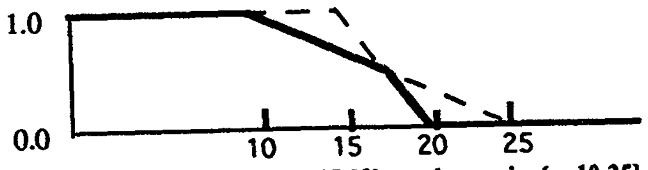


FIGURE 3
Trapezoidal Membership Functions

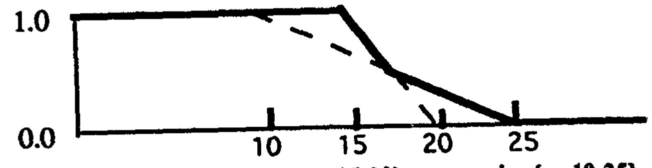
The prototype supports multiple fuzzy predicates during a single selection and combines the results of membership function calculations using standard fuzzy logic operators for fuzzy *and, or* and *not* operations.

Select * where name=smith and age is YOUNG and height is TALL

Due to the characteristics of the current datafilter, membership functions are limited to piecewise linear functions. The restriction is due to the lack of a multiply instruction in the VLSI datafilter. Our implementation uses repetitive addition to emulate a multiply instruction. The combination of multiple membership functions with overlapping domains can be used to approximate slightly more complex functions.



Select * where age is { ,, 15,20} and age is { ,,10,25}

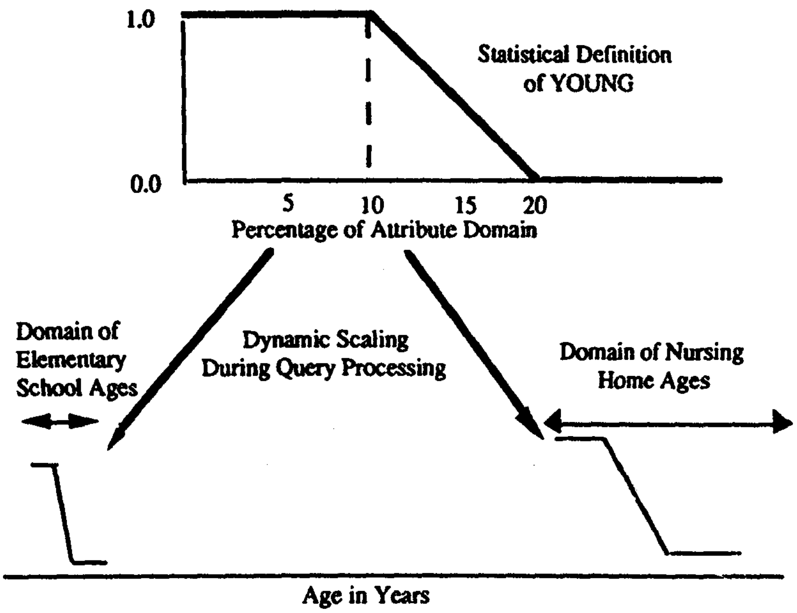


Select * where age is { ,,15,20} or age is { ,,10,25}

The selection process permits both hardware and combined hardware-software filtering. The VLSI datafilter has the responsibility of reducing the amount of information presented on the high bandwidth channel to an I/O bandwidth the downstream database processing environment can manage. Where possible, it is usually advantageous to complete this filtering operation in the VLSI datafilter. Where the complexity of the request exceeds the capability of the datafilter, a partial predicate, or an approximation can be used in the datafilter, and the downstream software can complete the predicate or apply a precise operation. For instance, we have approximated a distance function in the datafilter with an approximation $D_{xy} = |x| + |y|$. For some applications, this is sufficient. We use this distance calculation in a fuzzy *near* predicate. For those applications requiring a precise distance function, the approximation can yield a superset of the answer set, and downstream software can apply a Euclidean distance function to identify correct tuples or a correct ordering of tuples. This technique achieves greater flexibility and permits applications where the selection predicates exceed the capacity or primitives of the VLSI datafilter. Using this technique it is possible to approximate non-linear functions with piecewise linear functions, and subsequently apply the precise non-linear function in software outside the datafilter.

4. Dynamic Fuzzy Queries

One problem with a static predefinition of membership functions (i.e. YOUNG is less than age 25) is that the binding may make sense relative to the domain of the attribute in general (over all age groups), but for specific cases, may make no sense at all. For instance, suppose we were to apply the fuzzy predicate YOUNG as defined in Figure 1 to either elementary school children or nursing home adults. The definition is totally inappropriate. To partially overcome this shortcoming, we have implemented a dynamic fuzzy predicate which defines the membership function in terms of statistics and dynamically adjusts the function to the domain of the predicate. In this case YOUNG is defined in terms of percentiles of the domain space and interpreted as definitely young in the first 10th percentile, decreasing in membership value for the 10-20th percentile and not YOUNG beyond the 20th percentile. When applied to the domain of the predicate, the membership function is scaled appropriately as depicted in Figure 4. In the case of the current Datacycle prototype, the domain can be obtained by simply determining the maximum and minimum values of an attribute given additional predicate constraints (elementary school or nursing home). This can be accomplished by observing the data stream on a single cycle prior to the actual fuzzy selection. Using multiple filters or additional cycles, data distributions can be obtained if severe data skew is present and needs to be taken into consideration. The select statements indicated in Figure 4 show the extended SQL for the dynamically scaled requests. This approach does not create membership functions as in [Kam90], but rather, transforms "existing" membership functions to different populations.


1.0
0.0
Statistical Definition of YOUNG
5
10
15
20
Percentage of Attribute Domain
Domain of Elementary School Ages
Dynamic Scaling During Query Processing
Domain of Nursing Home Ages
Age in Years

```
select where age RIS YOUNG and group=elementary
select where age RIS YOUNG and group=nursing
```

Figure 4
Dynamic Fuzzy Query Processing

In the grammar, we use the term *RIS* (Relative IS) in place of the *IS* term to specify that a fuzzy term in the predicate is to be scaled. The use of the RIS operator cleanly differentiates membership functions as statistical functions. Thus, the predicate "age is YOUNG" would not be scaled and require the membership function be defined for a specific range of ages, and "age RIS YOUNG" would dynamically adjust the membership function.

The ability to dynamically scale the membership function to the domain of an arbitrary subset of the data is a significant departure from approaches that depend on the static definition of the membership function. A single statistical definition for OLD can used to identify old people, old computers, old pets, or old data files, and the ages can be expressed in years, months, days, hours or any unit. This approach enables meaningful fuzzy query processing for a much larger set of applications, and reduces the amount of database specific knowledge required of a user.

5. Performance

We have generated a modest test database of 256,000 tuples (32 Megabytes) to exercise the fuzzy query functionality and populated several attributes with numeric data. The prototype can handle larger databases with linear degradation in response time (response time is largely a function of database broadcast cycle time). The largest database we can handle in our current research prototype is 128 Megabytes in a single Datacycle pump; sufficient storage to deal with 1 million 128 byte tuples because the storage requirements do not require additional space for indexes. Larger databases require partitioning the database over multiple pumps and utilizing multiple filters. This same approach is used to reduce the cycle time of databases to engineer the system for faster response time. We have demonstrated the on-the-fly calculation of two membership functions and the fuzzy logic necessary to perform intersection and union operations totally within the VLSI datafilter. However, the general processing algorithm is to project resulting membership values to software outside the VLSI where the results are sorted on-the-fly. A subsequent query is then executed against the datastream to retrieve the "best" n tuples. Currently we select up to the best 50 records because we can specify the selection of up to 27 specific records in a single cycle, and the max of 50 fits well in two cycles. The two cycle limit was chosen arbitrarily, the architecture can easily support the selection of any number of records.

Figure 5 provides response time and throughput results for fuzzy selection queries against the three different database sizes (256K, 128K and 64K tuples). Since the system scales throughput linearly with the addition of filtering subsystems, query processing systems can be constructed to deal with arbitrarily large numbers of fuzzy queries. Figure 5 indicates the query throughput (queries per second) for a *single* filter. For non-fuzzy queries we have attained query throughputs exceeding 25 queries per second for a single filter for 64,000 tuple databases. The results indicate that the response time is largely determined by the cycle time, as is the query throughput. Response times for small (64K tuple) databases can be as low as .5 seconds with the full ad-hoc capability. A second result is indicated by the two 64K tuple curves and shows that both response time and throughput can be largely independent of complexity. The two predicate curve represents processing a selection based on two fuzzy membership functions operating on different attributes (SELECT WHERE NAME=LEE AND HEIGHT IS TALL AND AGE IS YOUNG) and the combination of their results with fuzzy logic operators. Notice that at low concurrency (1-3 queries per second), the response time is nearly identical to the single attribute predicate. The response time represents the time necessary to receive an extended SQL request, parse and compile it, load the datafilter instruction buffer, evaluate every record in the database, sort the resulting possibility values, select individual records on a subsequent cycle, move records into a private workspace (buffer) in the data management process and notify the application of number of records selected. The timing of subsequent record fetch operations is not included. Relative (RIS) query processing adds an additional broadcast cycle to the processing to collect statistics for scaling and marginally impacts performance.

These results, coupled with our experience with non-fuzzy query processing lead us to the conclusion that with multiple filters and multiple broadcast streams, we can achieve sub-second response time for reasonably large databases (100s MByte to Gigabyte), complex queries, and high levels of query throughput (100s per second) in a database environment that can be shared by many diverse applications, including ad-hoc fuzzy queries.

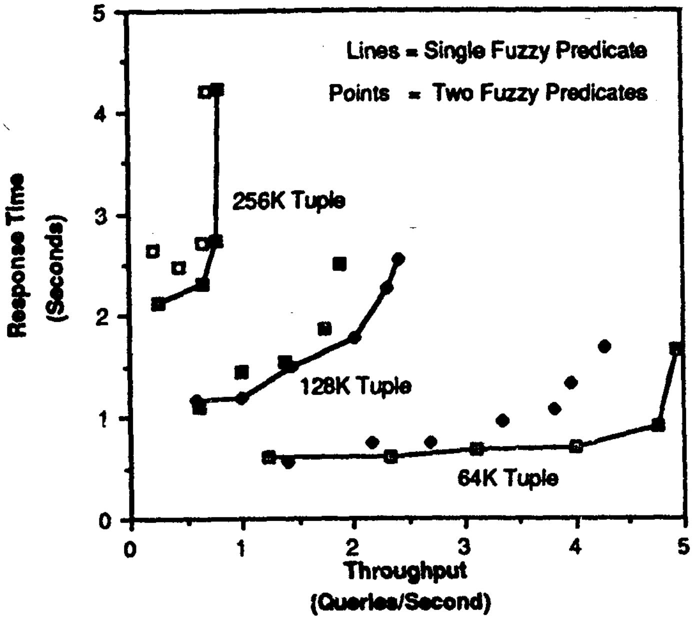


Figure 5
Fuzzy Query Performance - Single Filter
Select * where Name=Lee and height is tall

6. Future Work

Our work to date has centered around prototyping basic query funcuionality and implementing the dynamic fuzzy query capability. The full database search and content-addressability characteristics of the Datacycle architecture make it particularly attractive for a number of further extensions.

Multi-dimensional membership functions

Membership functions involving more that one attribute can be dealt with efficiently as a small change to the system since the values of all the tuple's attributes are available during the run-time evaluation of a membership function. Thus planar surfaces defined as $Z = c_1X + c_2Y + c_3$ are a possible alternative to lattice functions. Using multiple datafilters, multiple intersecting planes such as those depicted in Figure 6 are possible. We are particularly interested in spatial and directional issues such as *north* and the combination of direction and distance.

Hedges

A mechanism for modifying membership functions with standard hedge [Zad72] terms as *very* and *somewhat* needs to be addressed. These operators typically involve applying non-linear functions like the square or square-root of a membership value. These operators may be approximated using piecewise linear functions.

Concurrency Control in a Fuzzy Transaction Processing Environment

The Datacycle architecture and research prototype includes the implementation of a full transaction model to guarantee database consistency and query correctness in the face of concurrent transaction execution. The implementation includes optimistic concurrency control and a predicate based conflict detection algorithm that may prove advantageous in identifying conflicts between fuzzy transactions where standard record based locking schemes may be inappropriate. Relaxing strict concurrency control serialization requirements by using various forms of flexible transactions is an active area of current research in the database community. The notion of low and high conflicts in read/writeset intersections could utilize a mechanism to evaluate set intersections in a fuzzy domain. One approach can be the use of a threshold to limit intersections to only those that exceed a certain level of "conflict". In the Datacycle concurrency control mechanism, this could easily be implemented for transactions involving readsets that include fuzzy predicates by re-executing readsets against the log of database change activity coupled with a threshold set to a particular conflict level. Thus, only a subset of the conflicts would be identified during the optimistic concurrency control conflict detection phase, and only those that represent high degrees of conflict.

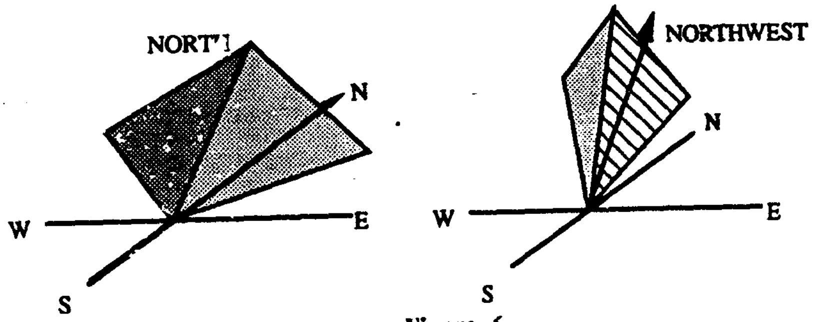


Figure 6
Two Dimensional membership Functions
North = F(Δlongitude, Δlatitude)

Complex Non-Linear Membership Functions

We are currently investigating the use of high performance digital signal processors (DSPs) for use instead of the custom VLSI datafilter. These processors include support for floating point and multiply instructions in as little as 40 nanoseconds. This is especially important since the current VLSI datafilter prototype lacks a multiply instruction. The use of DSPs could permit a significant increase in the complexity of membership functions that are executed on-the-fly.

Learning

We have speculated that the combination of on-the-fly membership function execution and membership function definition at the query grammar level provide the primitives for an adaptive feedback mechanism and eventually learning.

7. Conclusion

The combination of the Datacycle architecture's full database broadcast and efficient filtering can be used for both simple and complex database selection operations. This flexibility permits various applications to share a common database while requiring very different views of the data (full content-addressability), or permit a user to process searches beyond the capabilities of current database management systems. The work reported in this paper resulted in a fuzzy query capability in a high volume query processing environment. The major contributions of this work include a scalable query environment for fuzzy queries against small to medium size databases (low-order gigabyte), the use of on-the-fly membership function execution that permits ad-hoc fuzzy queries, and the definition and implementation of dynamic fuzzy queries that adjust a statistical membership function to the attribute domain.

Acknowledgments

We would like to thank other members of the Datacycle prototype research team including Thomas Bowen, Munir Cochinwala, Kasey Lee, and John Raitz, and to Bill Buckles for his advice and assistance while we were identifying and pursuing fuzzy database issues.

References

[Bosc88] Bosc, P., Galibourg, M., Hamon, G., "Fuzzy Querying with SQL: Extensions and Implementation Aspects", *Fuzzy Sets & Systems*, Vol 28, pp 333-349, 1988.

[Bow91] Bowen, T.F., Gopal, Herman, G.E., Mansfield, W.H., "A Scalable Database Architecture for Network Services", *IEEE Communications Magazine*, January 1991, pp 52-59.

[Bow92] Bowen, T.F., Fleischman, R.M., Herman, G.E., Mansfield, W.H., "The Datacycle Architecture and Research Prototype: A Quantitative Analysis", *Research Issues In Data Engineering*, February 1992

[Buc83] Buckles, B. P., Petry, F. E., "Query Languages for Fuzzy Databases, *Management Decision Support Systems*, pp 241-252, 1983

[Buc87] Buckles, B. P., Petry, F. E., "Generalized Database and Information Systems", *Analysis of Fuzzy Information*, Vol 2, pp 177-201, 1987.

[Codd70] Codd, E.F., "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM*, June 1970

[Dub80] Dubois, D, Prade, H. "*Fuzzy Sets and Systems: Theory and Applications*", Academic Press, New York, 1980

[East87] Eastman, C. "Approaches to Approximate Retrieval in Database Management Systems", *NAFIPS 87: Proceedings of the North American Fuzzy Information Processing Society Workshop*, May 1987

[Gala91] Gala,S., Chawala, D., Eastman, C., "Combining Fuzzy and Nonfuzzy Approximate Retrieval in a Database Management System", *NAFIPS 91 Proceedings of the North American Fuzzy Information Processing Society Workshop* , May 1991

[Her87] Herman, G.E., Gopal, G., Lee, K.C., Weinrib, A., "The Datacycle Architecture for Very High Throughput Database Systems", *Proceedings of the ACM SIGMOD*, 1987.

[Kac89] Kacprzyk, J., Zadronsky, S., Ziolkowski, A., "FQUERY III+: A Human Consistent Database Querying System Based on Fuzzy Logic with Linguistic Quantifiers", *Information Systems*, Vol 14, No. 6, pp 443-453, 1989.

[Kam90] Kamel, M., Hadfield, B., Ismail, M., "Fuzzy Query Processing using Clustering Techniques", *Information Processing & Management*, Vol 26 No. 2, pp 279-293, 1990.

[Lee91] Lee, K. C., Matoba, T., Herman, G. E., Mansfield, W.H., "A Rapid Turnaround Design of a High Speed VLSI Search Processor", *Integration: the VLSI Journal*, Vol. 10, pp. 319-337, 1991

[Lee91] Lee, K. C., Matoba, T., Mak, V., "VLSI Accelerators for Large Database Systems", *IEEE Micro*, December 1991

[Prad87] Prade, H., Testemale, C.,"Representation of Soft Constraints and Fuzzy Attribute Values by means of Possibility Distributions in Databases", *Analysis of Fuzzy Information*, Vol 2, pp 213-229, 1987

[Tah77] Tahani, Voliollah, "A Conceptual Framework for Fuzzy Query Processing - A Step toward Very Intelligent Database Systems", *Information Processing & Management*, Vol 13, pp 289-303, 1977.

[Uma83] Umano, M., "Retrieval from Fuzzy Database by Fuzzy Relational Algebra", *Proc. IFAC Conf. on Fuzzy Information, Knowledge Representation and Decision Analysis*, pp 1-16, 1983.

[Zad65] Zadeh, L.A., "Fuzzy Sets", *Information and Control*, Vol 8, pp 338-353, 1965.

[Zad72] Zadeh, L.A., "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges", *Journal of Cybernetics*, Vol 2, pp. 4-34, 1972.

[Zem84] Zemankova, M., Kandel, A., "A Fuzzy Relational Database - a Key to Expert Systems, Interdisciplinary Systems Research, Verlag TNV, Rheinland, 1984

[Zem85] Zemankova, M., Kandel, A., "Implementing Imprecision in Information Systems", *Information Sciences* 37(1,2,3), pages 107-141

N93

29572

UNCLAS

GENETIC ALGORITHMS IN ADAPTIVE FUZZY CONTROL

By

C. Lucas Karr and Tony R. Harper
U.S. Bureau of Mines, Tuscaloosa Research Center
P.O. Box L, University of Alabama Campus
Tuscaloosa, AL 35486-9777

ABSTRACT

Researchers at the U.S. Bureau of Mines have developed adaptive process control systems in which genetic algorithms (GAs) are used to augment fuzzy logic controllers (FLCs). GAs are search algorithms that rapidly locate near-optimum solutions to a wide spectrum of problems by modeling the search procedures of natural genetics. FLCs are rule based systems that efficiently manipulate a problem environment by modeling the "rule-of-thumb" strategy used in human decision making. Together, GAs and FLCs possess the capabilities necessary to produce powerful, efficient, and robust adaptive control systems. To perform efficiently, such control systems require a ***control element*** to manipulate the problem environment, an ***analysis element*** to recognize changes in the problem environment, and a ***learning element*** to adjust fuzzy membership functions in response to the changes in the problem environment. Details of an overall adaptive control system are discussed. A specific computer-simulated chemical system is used to demonstrate the ideas presented.

INTRODUCTION

The need for efficient process control has never been more important than it is today because of economic stresses forced on industry by processes of increased complexity and by intense competition in a world market. No industry is immune to the cost savings necessary to remain competitive; even traditional industries such as mineral processing (Kelly and Spottiswood, 1982), chemical engineering (Fogler, 1986), and wastewater treatment (Gottinger, 1991) have been forced to implement cost-cutting measures. Cost-cutting generally requires the implementation of emerging techniques that are often more complex than established procedures. The new processes that result are often characterized by rapidly changing process dynamics. Such systems prove difficult to control with conventional strategies, because these strategies lack an effective means of adapting to change. Furthermore, the mathematical tools employed for process control can be unduly complex even for simple systems.

In order to accommodate changing process dynamics yet avoid sluggish response times, adaptive control systems must alter their control strategies according to the current state of the process. Modern technology in the form of high-speed computers and artificial intelligence (AI) has opened the door for the development of control systems that adopt the approach to adaptive control used by humans, and perform more efficiently and with more flexibility than conventional control systems. Two powerful tools for adaptive control that have emerged from the field of AI are fuzzy logic (Zadeh, 1973) and genetic algorithms (GAs) (Goldberg, 1989).

The U.S. Bureau of Mines has developed an approach to the design of adaptive control systems, based on GAs and FLCs, that is effective in problem environments with rapidly changing dynamics. Additionally, the resulting controllers include a mechanism for handling inadequate feedback about the state or condition of the problem environment. Such controllers are more suitable than past control systems for recognizing, quantifying, and adapting to changes in the problem environment.

The adaptive control systems developed at the Bureau of Mines consist of a ***control element*** to manipulate the problem environment, an ***analysis element*** to recognize changes in the problem environment, and a ***learning element*** to adjust to the changes in the problem environment. Each component employs a GA, a FLC, or both, and each is described in this paper. A particular problem environment, a computer-simulated chemical system, serves as a forum for presenting the details of an adaptive controller being developed by the Bureau. Preliminary results are presented to demonstrate the effectiveness of a GA-based FLC for each of the three individual elements.

PROBLEM ENVIRONMENT

In this section, a computer-simulated chemical system is introduced to serve as a forum for presenting the details of a stand-alone, comprehensive, adaptive controller being developed at the U.S. Bureau of Mines; emphasis is on the method not the application. The chemical system consists of a continuous stirred tank reactor in which ammonia and formaldehyde are mixed to produce hexamine and water. Since the reaction is exothermic, a heat exchanger is included to limit the temperature in the reactor. A schematic of the physical system is shown in Figure 1.

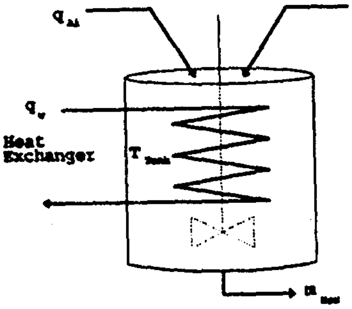


Figure 1.--A schematic of the hexamine system.

A mathematical model used in this research employs the approach described by Kermode and Stevens (1965). Specifically, the system is modelled with the following set of equations:

Energy Balance

$$q_A \rho C_p T_{Ai} + q_F \rho C_p T_{Fi} - (q_A + q_F) \rho C_p T_{tank} + r(-\Delta H)V - UA\Delta T_m = V \rho C_p \frac{dT}{dt}$$

Mass Balances

$$q_A C_{Ai} - (q_A + q_F) C_A - rV = V \frac{dC_A}{dt}$$

$$q_F C_{Fi} - (q_A + q_F) C_F - 1.5rV = V \frac{dC_F}{dt}$$

Heat of reaction

$$-\Delta H = 16610 + 121(T_{tank} - 293.2)$$

Rate of reaction

$$r = kC_A C_F^2$$

where q represents the volumetric flow rates (l/s), C is the concentration (moles/l), r is the rate of reaction (moles of ammonia/l s), V is the volume of the reactor (tank) (l), T is temperature (°K), ΔH is the heat of reaction (cal/gm mole), U is the heat transfer coefficient (cal/cm^2 °K s), ΔT_m is the mean temperature difference for heat transfer in the heat exchanger (and is a function of the volumetric flow rate of water through the heat exchanger, q_w), and k is the rate of reaction constant (l^2/mole^2 s) given by:

$$k = 1420 * e^{\frac{-3090}{T_{tank}}}$$

and the subscripts (A and F) indicate the ammonia and formaldehyde whereas the subscript i represents material entering the reactor. The assumptions associated with this model include perfect mixing in the reactor, no heat losses, all physical properties the same as water, and a third-order, irreversible reaction.

A reactor having a volume of 92.4 l was simulated. The inflows of ammonia and formaldehyde, respectively, were allowed to reach maximum values of 1.885 l/s, while the maximum flow rate of the heat exchanger was 1.2 l/s. The objective of the control problem is two-fold: (1) to develop a FLC capable of maintaining a desired reactor temperature in response to changes in the flow rate of formaldehyde and (2) to maximize the production of hexamine while minimizing the waste in the amount of reactants used. The amount of water produced was deemed inconsequential to the control strategy. In this research, the desired reactor temperature is 315.0 °K. Furthermore, a constraint is placed on the amount the valves controlling the inflow of ammonia can be opened or closed during a given time step. The maximum rate at which the flow of ammonia can be changed is 0.1885 l/s/s. This constraint is enforced to limit transients in the system.

The hexamine system, as it has been described to this point, provides a challenging control problem, due mainly to the nonlinearity present in the rate of reaction. It is a non-trivial task to maintain the temperature in the reactor for various forcing functions (as defined by the rate at which the formaldehyde enters the reactor), much less to ensure the process proceeds efficiently (maximum hexamine production with minimal waste in ammonia and formaldehyde). However, yet another complication is now introduced: the concentration of the reactants (the ammonia and the formaldehyde) can be altered randomly. Furthermore, there is no mechanism in place for providing the controller with feedback concerning the nature of these changes. Thus, an efficient control system must be able to recognize when the hexamine system has been altered (when the concentration of the reactants are changed), it must be able to determine the new values of the concentrations, and it must be able to alter its control strategy in response to the changes; an adaptive control system is needed.

STRUCTURE OF THE ADAPTIVE CONTROLLER

Figure 2 shows a schematic of the Bureau's adaptive control system. The heart of this control system is the loop consisting of the control element and the problem environment. The control element receives information from sensors in the problem environment concerning the status of the *condition variables*, i.e., q_A, q_F, q_w, and T_{tank}. It then computes a desirable state for a set of *action variables*, i.e., flow rate of ammonia (q_A) and flow rate of water through the heat exchanger (q_w). These changes in the action variables force the problem environment toward the setpoint (T_{tank} = 315.0° K. This is the basic approach adopted for the design of virtually any closed loop control system, and in and of itself includes no mechanism for adaptive control.

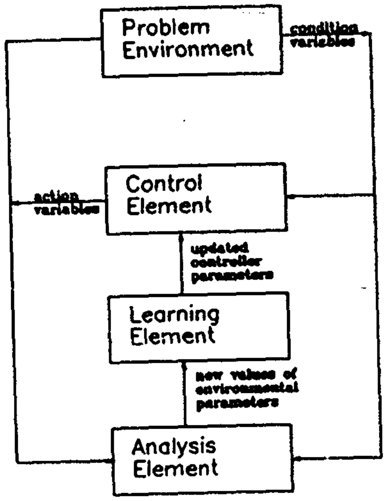


Figure 2.--Structure of the adaptive control system.

The adaptive capabilities of the system shown in Fig. 2 are due to the analysis and learning elements. In general, the analysis element must recognize when a change in the problem environment has occurred. A "change," as it is used here, consists of a change to the concentration of either of the reactants. The analysis element uses information concerning the condition and action variables over some finite time period to recognize changes in the environment and to compute the new performance characteristics associated with these changes.

The new environment (the problem environment with the altered parameters) can pose many difficulties for the control element, because the control element is no longer manipulating the environment for which it was designed. Therefore, the algorithm that drives the control element must be altered. As shown in the schematic of Fig. 2, this task is accomplished by the learning element. The most efficient approach for the learning element to use to alter the control element is to utilize information concerning the past performance of the control system. The strategy used by the control, analysis, and learning elements of the stand-alone, comprehensive adaptive controller being developed by the U.S. Bureau of Mines is provided in the following sections.

Control Element

The control element receives feedback from the hexamine system, and based on the current state of q_A, q_F, q_w, and T_{tank}, must prescribe appropriate values of q_A and q_w. Any of a number of closed-loop controllers could be used for this element. However, because of the flexibility needed in the control system as a whole, a FLC is employed. Like conventional rule-based systems, FLCs use a set of production rules which are of the form:

IF {*condition*} THEN {*action*}

to arrive at appropriate control actions. The left-hand-side of the rules (the *condition* side) consists of combinations of the controlled variables (q_A, q_F, q_w, and T_{tank}); the right-hand-side of the rules (the *action* side) consists of combinations of the manipulated variables (q_A and q_w). Unlike conventional expert systems, FLCs use rules that utilize fuzzy terms like those appearing in human rules-of-thumb. For example, a valid rule for a FLC used to manipulate the hexamine system is:

IF {q_A is VH and q_F is VL and q_w is L and T_{tank} is VH}
THEN {q_A is NB and q_w is PB}.

The fuzzy terms are subjective; they mean different things to different "experts," and can mean different things in varying situations. Fuzzy terms are assigned concrete meaning via fuzzy membership functions (Zadeh, 1973).

The membership functions used in the control element to describe ammonia flow rate appear in Fig. 3. (As will be seen shortly, the learning element is capable of changing these membership functions in response to changes in the problem environment.) These membership functions are used in conjunction with the rule set to prescribe single, crisp values of the action variables (q_A and q_w). Unlike conventional expert systems, FLCs allow for the enactment of more than one rule at any given time. The single crisp action is computed using a weighted averaging technique that incorporates both a *min-max* operator and the *center-of-area* method (Karr, 1991). The following fuzzy terms were used, and therefore "defined" with membership functions, to describe the significant variables in the hexamine system:

q_A Very Low (VL), Low (L), Medium (M), High (H), Very High (VH)
q_F Low (L), Medium (M), High (H), Very High (VH)
q_w Low (L), Medium (M), High (H)
T_{tank} Very Low (VL), Low (L), Medium (M), High (H), Very High (VH)
q_w Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB)
q_A Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB).

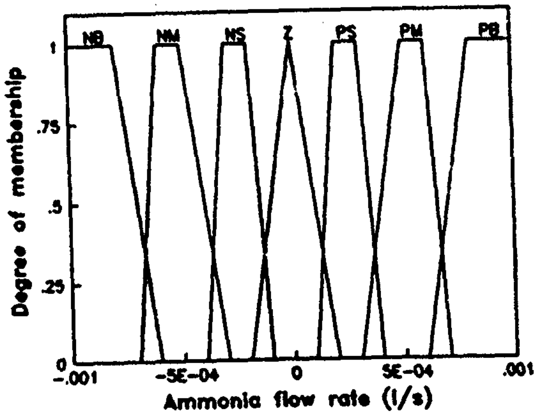


Figure 3.--Fuzzy membership functions for the flow rate of ammonia.

An effective FLC for manipulating the hexamine system can be written that contains 300 rules, *if* the random changes to the concentrations of the reactants are neglected. The 300 rules are necessary because there are five fuzzy terms describing T_{tank}, five fuzzy terms describing q_A, four fuzzy terms describing q_F, and three fuzzy terms describing q_w (5*5*4*3=300 rules to describe all possible combinations that could exist in the hexamine system as described by the fuzzy terms represented by the membership functions selected). Now, the rules selected for the control element are certainly inadequate to control the full-scale hexamine system; the one that includes the changing concentrations. However, the performance of a FLC can be dramatically altered by changing the membership functions. This is equivalent to changing the definition of the terms used to describe the variables being considered by the controller. As will be seen shortly, GAs are powerful tools capable of rapidly locating efficient fuzzy membership functions that allow the controller to accommodate changes in the concentrations of the reactants.

Analysis Element

The analysis element recognizes changes in parameters associated with the problem environment not taken into account by the rules used in the control element. In the hexamine system, these parameters are the concentration of the two reactants. Changes to the concentrations dramatically alter the way in which the hexamine system responds to control actions, thus forming a new problem environment requiring an altered control strategy. Recall that the FLC used for the control element presented includes none of these parameters in its 300 rules. Therefore, some mechanism for altering the prescribed actions must be included in the control system. But before the control element can be altered, the control system must recognize that the problem environment has changed, and compute the nature and magnitude of the changes.

The analysis element recognizes changes in the system parameters by comparing the response of the system being controlled to the response of a model of the hexamine system. In general, recognizing changes in the parameters associated with the problem environment requires the control system to store information concerning the past performance of the problem environment. This information is most effectively acquired through either a data base or a computer model. Storing such an extensive data base can be cumbersome and requires extensive computer memory. Fortunately, the dynamics of the hexamine system are well understood. In the approach adopted here, a computer model predicts the response of the hexamine system being controlled. This predicted response is compared to the response of the system being controlled. When the two responses differ by a threshold amount over a finite period of time, the hexamine system is considered to have been altered.

When the above approach is adopted, the problem of computing the new system parameters becomes a curve fitting problem (Karr, Stanley, and Scheiner, 1991). The parameters associated with the computer model produce a particular response to changes in the action variables. The parameters must be selected so that the response of the model matches the response of the problem environment.

An analysis element has been forged in which a GA is used to compute the values of the parameters associated with the hexamine system. When employing a GA in a search problem, there are basically two decisions that must be made: (1) how to code the parameters as bit strings and (2) how to evaluate the merit of each string (the fitness function must be defined). The GA used in the analysis element employs concatenated, mapped, unsigned binary coding (Karr and Gentry, 1992). The bit-strings produced by this coding strategy were of length 16: the first ** bits of the strings were used to represent the concentration of the ammonia and the second 8 bits were used to represent the concentration of the formaldehyde. The 8 bits associated with each individual parameter were read as a binary number, converted to decimal numbers (000 = 0, 001 = 1, 010 = 2, 011 = 3, etc.,), and mapped between minimum and maximum values according to the following:

$$C = C_{min} + \frac{b}{(2^m - 1)} (C_{max} - C_{min}) \qquad (7)$$

where C is the value of the parameter in question, *b* is the binary value, *m* is the number of bits used to represent the particular parameter (8), and C_{min} and C_{max} are minimum and maximum values associated with each parameter that is being coded.

A fitness function has been employed that represents the quality of each bit-string; it provides a quantitative evaluation of how accurately the response of a model using the new model parameters matches the response of the system being controlled. The fitness function used in this application is:

$$f = \sum_{t=0s}^{t=500s} |T_{tank_{model}} - T_{tank_{actual}}|. \qquad (8)$$

With this definition of the fitness function, the problem becomes a minimization problem: the GA must minimize f, which as it has been defined, represents the difference between the response predicted by the model and the response of the system being controlled.

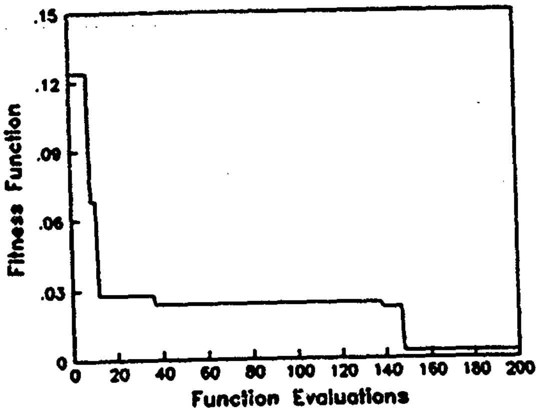


Figure 4.--A GA is able to compute the concentrations of the reactants.

Figure 4 demonstrates the ability of a GA to select the appropriate parameters associated with the problem environment. A GA is able to reduce the difference between the response of the hexamine system being controlled, and the response of the hexamine system predicted by the model virtually to zero after only 150 function evaluations. Once new parameters (and thus the new response characteristics of the problem environment) have been determined, the adaptive element must alter the control element.

Learning Element

The learning element alters the control element in response to changes in the problem environment. It does so by altering the membership functions employed by the FLC of the control element. Since none of the ranuomly altered parameters appear in the FLC rule set, the only way to account for these conditions (outside of completely revamping the system) is to alter the membership functions employed by the FLC. These alterations consist of changing both the position and location of the trapezoids used to define the fuzzy terms.

Altering the membership functions (the definition of the fuzzy terms in the rule set) is consistent with the way humans control complex systems. Quite often, the rules-of-thumb humans use to manipulate a problem environment remain the same despite even dramatic changes to that environment; only the conditions under which the rules are applied are altered. This is basically the approach that is being taken when the fuzzy membership functions are altered.

The U.S. Bureau of Mines uses a GA to alter the membership functions associated with FLCs, and this technique has been well documented (Karr, 1991). A learning element that utilizes a GA to locate high-efficiency membership functions for the dynamic hexamine system has been designed and implemented.

The performance of a control system that uses a GA to alter the membership functions of its control element is demonstrated for the situation in which the concentrations of both reactants are altered. Figure 5 compares the performance of the adaptive control system (one that changes its membership functions in response to changes in the system parameters) to a non-adaptive control system (one that ignores the changes in the system parameters). In this figure, the concentrations of both reactants have been altered 510 seconds into the simulation. In this case, not only is the adaptive controller able to better maintain the desired tank temperature, but it also prescribes control actions that allow for the production of more hexamine.

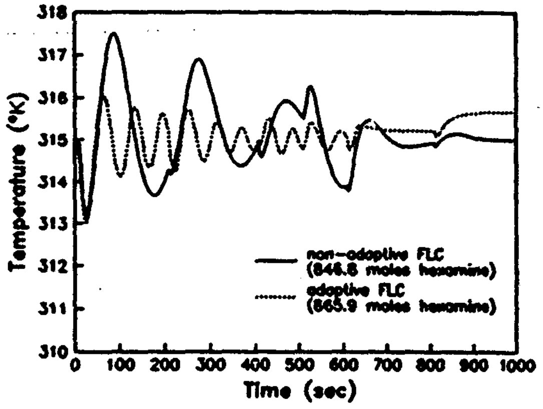


Figure 5.--The adaptive controller is much more efficient.

SUMMARY

Scientists at the U.S. Bureau of Mines have developed an AI-based strategy for adaptive process control. This strategy uses GAs to fashion three components necessary for a robust, comprehensive adaptive process control system: (1) a control element to manipulate the problem environment, (2) an analysis element to recognize changes in the problem environment, and (3) a learning element to adjust to changes in the problem environment. The application of this strategy to a computer-simulated hexamine system has been described.

REFERENCES

Fogler, H. S. (1986). *Elements of Chemical Reaction Engineering*. Prentice-Hall, Englewood Cliffs, NJ.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

Gottinger, W. W. (1991). *Economic Models and Applications of Solid Waste Management*. Gordon and Breach Science Publishers, New York, NY.

Karr, C. L. (1991). Genetic algorithms for fuzzy logic controllers. *AI Expert*, 6, 26-33.

Karr, C. L., Stanley, D. A., and Scheiner, B. J. (1991). *A Genetic Algorithm Applied to Least Squares Curve Fitting*. U.S. Bureau of Mines Report of Investigations No. 9339.

Kelly, E. G. and Spottiswood, D. J. (1982). *Introduction to Mineral Processing*. John Wiley & Sons, New York, NY.

Kermode, R. I., and Stevens, W. F. (1965). Experimental verification of the mathematical model for a continuous stirred-tank reactor. *The Canadian Journal of Chemical Engineering*, April, 68-72.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 28-44.

517-67
161857

p- 9

A Genetic Algorithms Approach for Altering the Membership Functions in Fuzzy Logic Controllers

Hana Shehadeh
LinCom Corporation
Houston, TX 77058
(713) 488-5700

Robert N. Lea
NASA/Johnson Space Center - FM7
Houston, TX 77058
(713) 483-8015

Abstract

Through previous work, a fuzzy control system was developed to perform translational and rotational control of a space vehicle. This problem was then re-examined to determine the effectiveness of genetic algorithms on fine tuning the controller. This paper explains the problems associated with the design of this fuzzy controller and offers a technique for tuning fuzzy logic controllers.

A fuzzy logic controller is a rule-based system that uses fuzzy linguistic variables to model human rule-of-thumb approaches to control actions within a given system. This "fuzzy expert system" features rules that direct the decision process and membership functions that convert the linguistic variables into the precise numeric values used for system control [7].

Defining the fuzzy membership functions is the most time consuming aspect of the controller design. One single change in the membership functions could significantly alter the performance of the controller. This membership function definition can be accomplished by using a trial and error technique to alter the membership functions creating a highly tuned controller. This approach can be time consuming and requires a great deal of knowledge from human experts.

In order to shorten development time, an iterative procedure for altering the membership functions to create a tuned set that used a minimal amount of fuel for velocity vector approach and station-keep maneuvers was developed. Genetic algorithms, a search technique used for optimization, was the method utilized to solve this problem.

1. PREVIOUS WORK

A previous effort in the Software Technology Lab at NASA/Johnson Space Center was directed towards creating a control system that reacted similarly to a pilot in the execution of rendezvous profiles [3], (discussed later in section 1.1). It was realized that a model of a man flying the space shuttle could be developed by allowing the system to process only information that the crew had available. Such a system would demonstrate feasibility of utilizing fuzzy controllers for automated rendezvous operations for future space missions.

1.1 SCENARIO

An automated six degree of freedom (6-DOF) fuzzy controller was developed [4] that performs four major translational control segments as shown in figure 1: 1) The approach to the target on the velocity vector, 2) the approach to the target on the radial vector, 3) target fly-around at a given range, and 4) station-keeping, where constant position relative to the target is maintained. The controller controls the closing rates and relative positions of the shuttle with respect to the stationary target. The velocity vector and radial vector approaches require range rate control along with maintaining the elevation and azimuth angles at zero. The station-keeping operation requires the range to be constant, the range rate to be near zero, and the azimuth and elevation angles to be zero. The fly-around operation from the velocity vector to the negative radial vector requires maintaining constant range, elevation and azimuth angles. In figure 1D, the elevation and azimuth angles are being maintained at zero.

Figure 1B shows the velocity vector approach, where the shuttle approaches from 400 feet to 50 feet from the target on the velocity vector, maintaining the target in the center of the field of view of the Crew Optical Alignment Sighting (COAS) device, as shown in figure 1A. Figure 1C shows the radial vector approach, where the shuttle approaches from 400 feet below the target to 50 feet below the target on the radial vector. Here again, the shuttle maintains the target in the center of the COAS field of view. Figure 1D shows the fly–around from the velocity vector axis to the negative radial vector axis, maintaining a range radius of 200 feet during the transition, and station keeping at 200 feet on the radial vector, maintaining position and attitude after reaching the desired radial vector position.

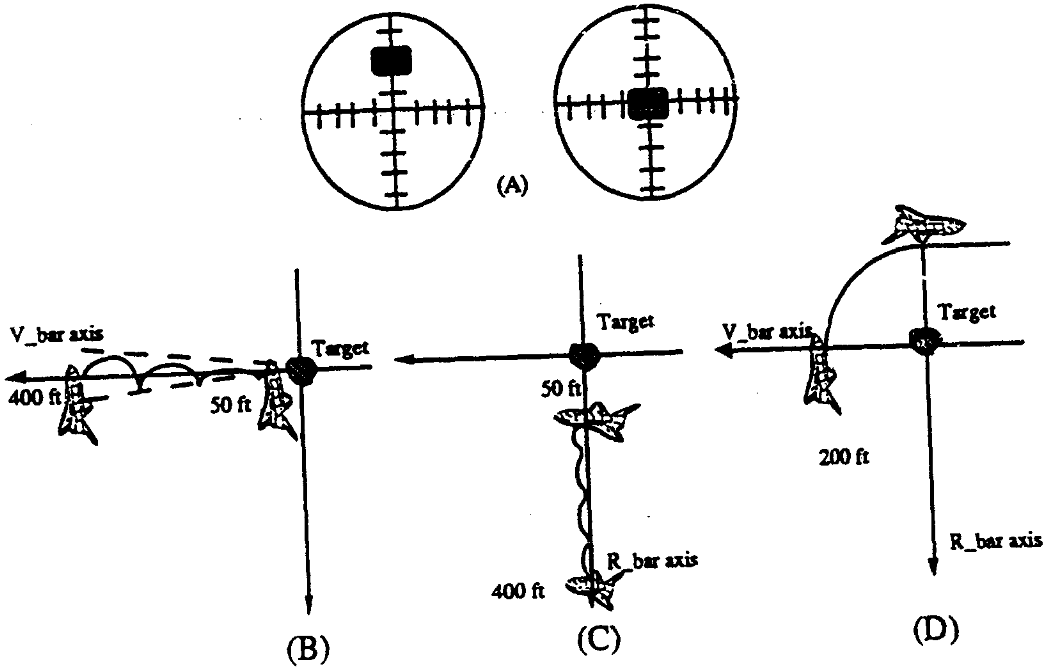


Figure 1: Automated Rendezvous System

2. CURRENT WORK

The translational fuzzy control system is used to generate hand controller commands so that the desired range and range rate are maintained during proximity operations. Typically, a shuttle pilot provides these inputs and controls the relative trajectory. Thus, the fuzzy logic based control system simulates the crew input via the translational hand controller [3].

In evaluating the performance of the fuzzy controller, fuel conservation was one of the main criteria used. The shuttle flights are pound for pound systems, i.e., for every pound of on-orbit fuel conserved, an equal amount of increase in the payload capacity results.

During the tuning of the controller, the membership functions were altered manually to improve the control strategy. Defining the fuzzy membership functions is the most time consuming aspect of the controller design. One single change in the membership functions significantly altered the performance of the controller. This membership function definition was accomplished by using trial and error techniques.

In order to shorten development time, an iterative procedure was developed for altering the membership functions to create a tuned set that reduced the amount of fuel for velocity vector approach and station-keep maneuvers. Genetic algorithms, a search technique used for optimization, is the method used to automate the fine-tuning of the membership functions in order to minimize fuel consumption. Figure 2 shows the final range parameters membership functions from the manually-tuned controller after extensive fine-tuning.

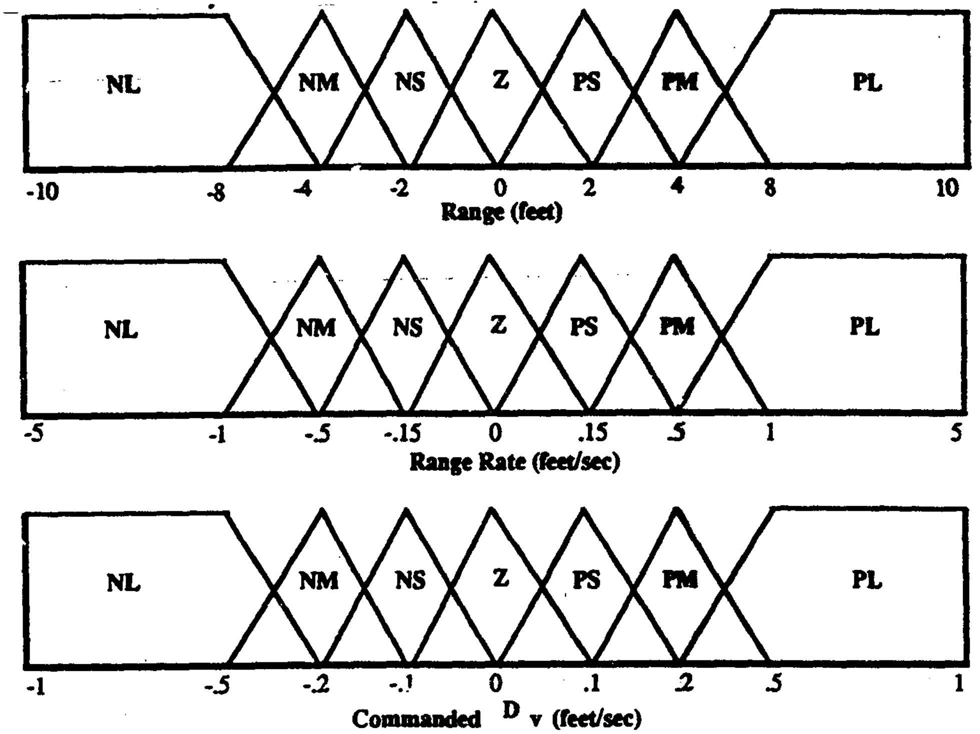


Figure 2: Range Parameters Membership Functions Set

Membership Functions Constraints

A chromosome string which consists of 38 points defined the range and range rate membership functions. The fitness function regulated the membership functions points to float along the universe of discourse within certain constraints. Figure 3 shows the labeled points. An example of the constraints algorithm placed on the individual points are as follows: The positive large (PL) vertex for range, labeled 3, is constrained to a value between the vertex of the positive medium membership function and the maximum value of the universe of discourse. The positive medium (PM) vertex for range, labeled 2, is constrained to a value between the vertex of the positive small (PS) membership function and the maximum value of the universe of discourse. The points labeled 4, 5, and 6 also follow this algorithm. The right leg of the positive medium membership function, labeled 13, is constrained to a value between the vertex of the positive medium membership function and the maximum value of the universe of discourse. The leg of the positive large membership function, labeled 12, is constrained to a value between the vertex of the zero membership function and the vertex of the positive large membership function. The points labeled 9, 10, 14, 15, 16, 17, and 18 also follow this algorithm. The points labeled 0, 7, 8, are fixed at zero and are not allowed to float. The membership functions for the range rate parameters are symmetric and follow the same algorithm for this approach.

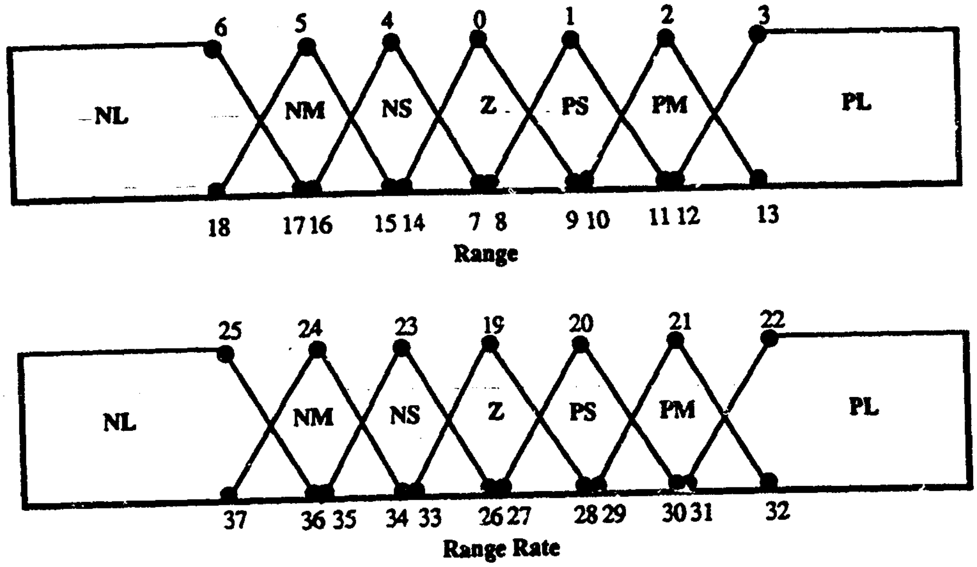


Figure 3: Selected Points

Figure 4 shows the functional flow of the genetic algorithm chromosome selection process. The process begins with random generation of a population of 50 strings (chromosomes), each of length 190 bits. Each normalized point was given a 5 bit resolution and decoded according to the constants described above. Each chromosome represents a possible solution to the problem of finding a set of highly efficient membership functions (with respect to fuel consumption). Each of these chromosomes which is a binary string, is decoded to yield the actual membership function parameters. Each chromosome is sent to the Orbital Operations Simulator (OOS) [6] (discussed in greater detail later in this paper), where simulated runs on the velocity vector and station-keeping are performed. The bit strings which represent the parameters of the search problem were then judged and assigned a score (a fitness-function value), that is a non-negative measure of relative worth, representing the degree to which they accomplish the goal of defining the high-performance fuzzy controller. The control parameters fuel and time, for both the approach maneuver and station keeping, were used to assign this score as shown in equation (1). Those chromosomes with a high fitness value measured by the equation are given a proportionately higher probability for selection during the representation phase.

$$Fitness = \frac{1}{(1 + (ApproachFuel * ApproachTime) + (StationKeepFuel * StationKeepTime))} \qquad (1)$$

Those parameters that were given the higher probability were placed in the new generation, a crossover and mutation process was performed on the new chromosomes and the process was re-iterated until a good solution was found.

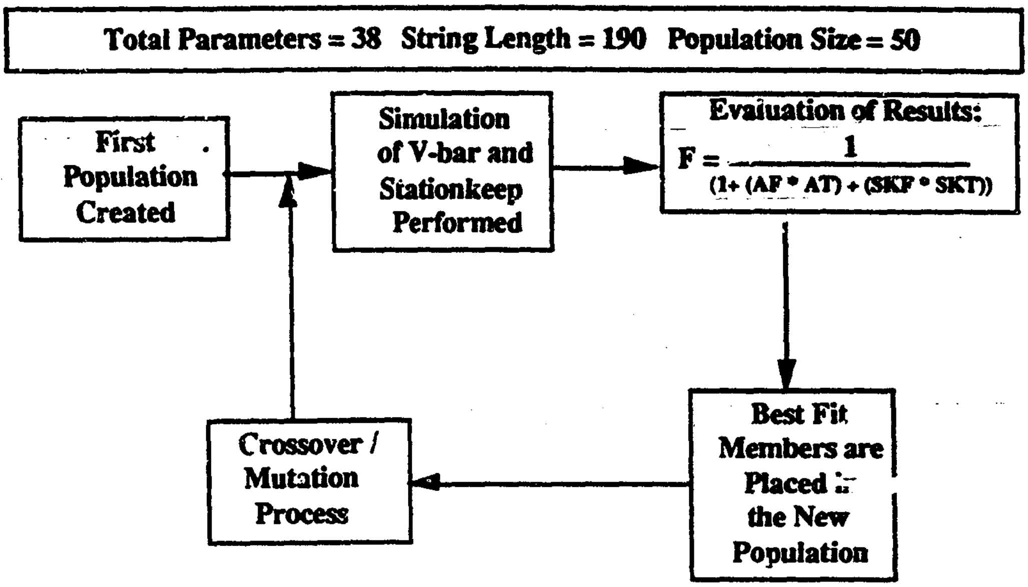

Total Parameters = 38 String Length = 190 Population Size = 50
First Population Created
Simulation of V-bar and Stationkeep Performed
Evaluation of Results:
F = 1 / (1+ (AF * AT) + (SKF * SKT))
Best Fit Members are Placed in the New Population
Crossover / Mutation Process

Figure 4: Flow Chart of the Selection Process

3. SPECIALIZED SOFTWARE TOOLS USED

Splicer

Splicer [1,2], a genetic algorithm tool designed for developing code for evaluating chromosomes was used in this project. The objectives of this approach were to evaluate the capabilities of genetic algorithms for the widespread use in automating the fine tuning of the fuzzy logic membership functions.

If successful this type of approach would be applicable in a variety of domains: e.g., robot path planning and job shop scheduling. Splicer is a flexible, generic tool that allows for:

- Implementing the basic genetic algorithms defined in the literature
- Defining the interfaces for and allowing users to develop interchangeable fitness modules
- Providing a graphic, event-driven user interface.

Splicer consists of a genetic algorithm kernel that comprises all functions necessary for the manipulation of populations including, the creation of populations and the population members, fitness scaling, and random number generation. It also provides representation libraries for binary strings and for permutations. The fitness modules are the only component of the Splicer system a user will be required to create or alter to solve a particular problem. Within a fitness routine a user can create a fitness (scoring) function and set the initial values for the control parameters. Splicer is available in X-Windows and Macintosh versions, as well as a generic C language command line version [1,2].

Orbital Operations Simulator (OOS)

For testing the 6-DOF controller, NASA's OOS was used with its graphics interface to the Iris workstation. The OOS is a high fidelity, multi-vehicle spacecraft operations simulation that provides 6-DOF equations of motion within an orbital environment including gravity gradient and aerodynamic drag.

The OOS has a high fidelity Space Shuttle model with the fuzzy 6-DOF controller and the required orbital environment math models. The OOS also has the capability of simulating mission timelines according to crew procedures. The 6-DOF fuzzy logic shuttle controller was implemented in the OOS environment and detailed simulation testing was used to evaluate its performance.

4. RESULTS

Table 1 shows the results of the two designs. A comparison was made between the outputs from the piloted simulation runs, the manually tuned controller outputs and the automatically fine-tuned controller outputs.

As can be seen, the performance of the fuzzy controller compares quite well to the piloted results. The manually tuned fuzzy controller outperforms the piloted control for rate of fuel usage on the velocity vector approach maneuver by .001 lbs/sec, and for station keep by .0133 lbs/sec.

Membership functions automatically tuned by the genetic algorithm produced results comparable to those achieved with the manually tuned fuzzy controller. The Fuzzy Genetic Algorithm controller used .002 lbs/sec more fuel than the manually tuned fuzzy controller but used .001 lbs/sec less than the fuel required for a piloted velocity vector approach maneuver. For station keep the Fuzzy Genetic Algorithm controller used .004 lbs/sec less fuel than the piloted results. However, the Fuzzy Genetic Algorithm used .0093 lbs/sec more fuel than the manually tuned fuzzy controller.

The above results from the genetic algorithm tuning approach are promising. It is anticipated that through future work, improved performance of the controller can be achieved by allowing the height of the vertex points to float as well as the positions in the domain.

Piloted Results

Maneuver Type	Distance (ft)	Rate of Fuel Used (fuel / time)
v_bar approach	400-50	.022 (lbs/sec)
Station Keep	@200	.02 (lbs/sec)

Manually Tuned Fuzzy Controller

Maneuver Type	Distance (ft)	Rate of Fuel Used (fuel / time)
v_bar approach	400-50	.023 (lbs/sec)
Station Keep	@200	.0067 (lbs/sec)

Automatically Tuned Fuzzy Controller

Maneuver Type	Distance (ft)	Rate of Fuel Used (fuel / time)
v_bar approach	400-50	.021 (lbs/sec)
Station Keep	@200	.016 (lbs/sec)

Table 1: Results Summary Table

The graph in figure 5 shows the relationship between the amount of fuel used versus the number of generations. As can be seen, several minima were found by the algorithm. The concern here was that the genetic algorithm never stabilized. Results that compared quite well to the manual were achieved, yet convergence to a minimal amount of fuel was not achieved.

C-3

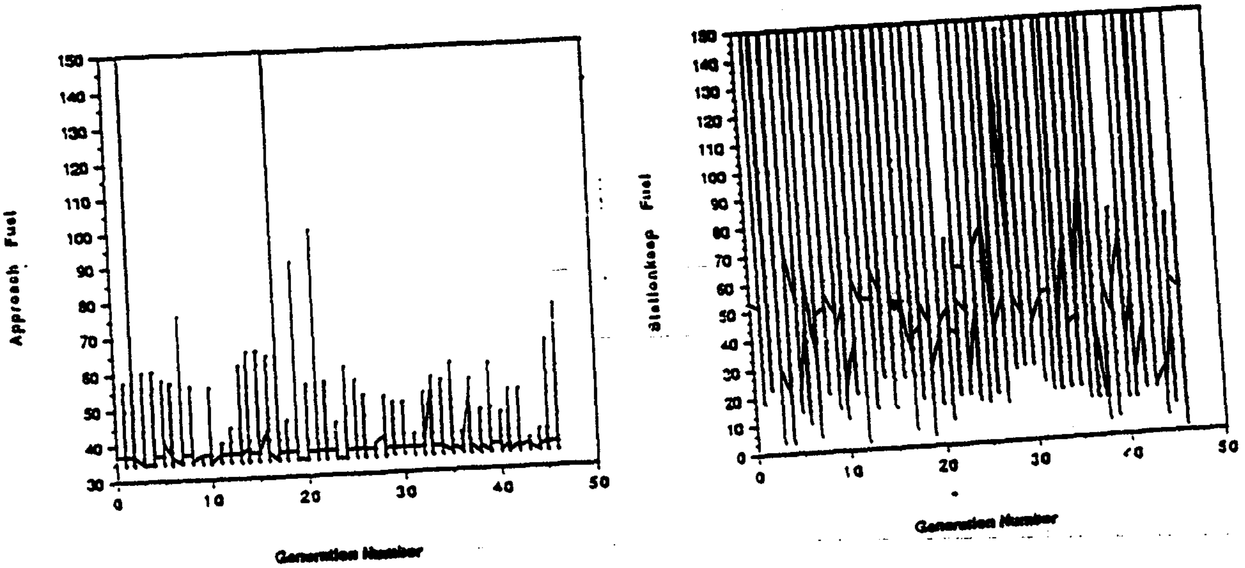
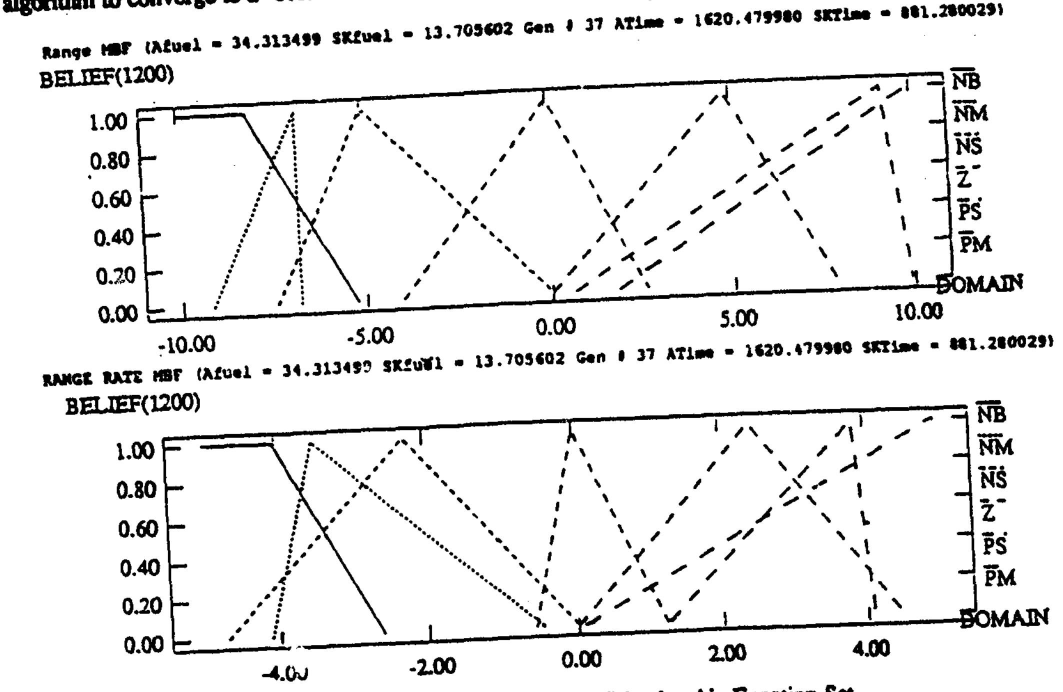


Figure 5: Fuel Used .vs. Generation Number

Figure 6 depicts the results after the thirty seventh generation. The results from this generation were chosen because this generation contained the last minima found by the algorithm after some stabilization. We found that more than one set of membership functions generated by the use of the genetic algorithms yielded identical performance, as far as, fuel usage and time. This suggested several possibilities to us as to the nature of our problem. First, factor, other then the ones we are changing through the genetic algorithm, may be having a significant effect on the control system. Also, it is possible that some of the items changed by the genetic algorithms have no effect on the controller (some membership functions are never used). These possibilities are discussed in more length in the issues and concerns section 5. However, since several sets of solutions seemed of the exact same value to the genetic algorithm it was very difficult, if not impossible, for the genetic algorithm to converge to a "best" solution.


Range MBF (Afuel = 34.313499 SKfuel = 13.705602 Gen # 37 ATime = 1620.479980 SKTime = 881.280029)

RANGE RATE MBF (Afuel = 34.313499 SKfuWl = 13.705602 Gen # 37 ATime = 1620.479980 SKTime = 881.280029)

Figure 6: The Genetic Algorithms/Membership Function Set

5. ISSUES/CONCERNS

The problem domain parameters defined for this approach are, Sampling Operator (TOURNAMENT), Population Size (50), and Mutation Probability (0.001).

A tournament type sampling operator is used to sample members of the population for mating. Sampling uses target sampling rates (generated by selection) to create a mating pool of members from the current population of size 50. Members may be chosen for mating multiple times or not at all, according to their target sampling rates. The mutation probability operator was set to .001[1,2].

These genetic operators were applied to the members of the population, and their strings, while the genetic algorithms were running.

As a feasibility study, the population size of fifty was chosen to fine-tune the membership functions. Considering the size of the chromosome string, a larger population may have been beneficial, however, the complexity of the evaluation process resulted in an evaluation time of several minutes per population member. Due to this computation time, a population size greater than fifty was prohibitive. Now that feasibility has been ascertained, a larger population size would be more appropriate to make the various runs on the various test cases.

For the test cases performed in this paper, the orbiter's starting position was at 400 feet from the target on the velocity vector. It is possible the test cases used for evaluation may have exercised only a portion of the control system. More test cases where the starting position of the orbiter is randomly initialized would have given a more accurate evaluation of the genetic algorithm's effectiveness.

Finally, the orbiter's starting position was always 400 feet from the target on the velocity vector, only three out of the seven sets of membership functions changed by the genetic algorithms for each of the parameters had an effect on the controller. The controller was controlled by the NS, Z and PS membership functions. The membership functions NL, NM, PM, and PL were never used. Having random starting positions of the orbiter would then require the use of all seven of the membership functions.

It is interesting to note that all three approaches (piloted, manually tuned, and fuzzy genetic algorithm) produced the same results for the velocity vector approach (~ 0.02 lbs/sec). A possible reason for this is that the controller is also controlled with breaking gates. When approaching a target, the orbiter adheres to a defined speed limit which is a function of the distance to the target. These range dependent rates are called "breaking gates" and are shown in figure 7. Outside of 400 feet the approach speed is 0.4 ft/sec. At 300 feet the allowable approach rate drops to 0.3 ft/sec. The 0.2 ft/sec rate is maintained from 200 feet to approach termination. Since the path taken on the approach is constrained, the results of all three approaches maybe the minimal fuel usage possible.

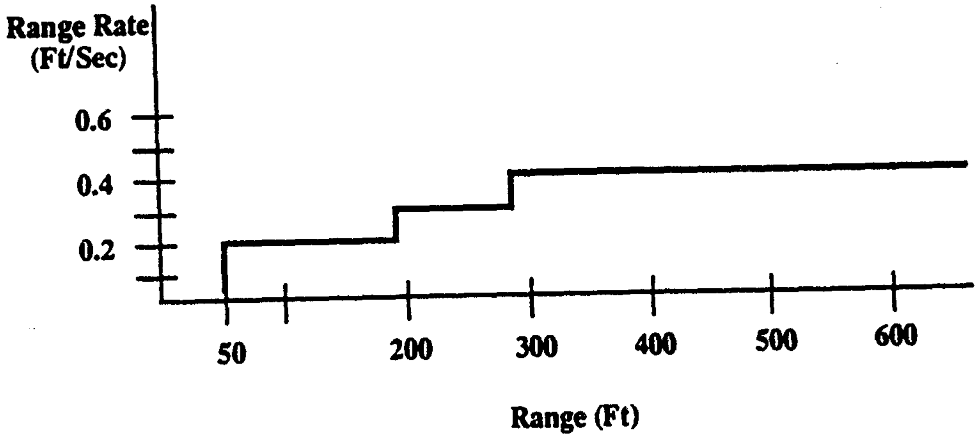


Figure 7: Breaking Gates Approach

6. SUMMARY

We have demonstrated the use of genetic algorithms to automate the fine tuning of fuzzy logic membership functions for a spacecraft proximity operations controller. The complexity of the problem and the resulting computational intensity of the genetic algorithm population member evaluations did place some constraints on our implementation of the problem. However, a solution comparable to highly trained pilots and a manually fine tuned controller was obtained in a reasonable amount of time. Genetic algorithms show a viable potential for the automatic fine–tuning of fuzzy logic based control systems.

Acknowledgments

The authors would like to thank Steven Bayer of MITRE Corporation, Jeff Hoblit of LinCom Corporation, Yashvant Jani of Togai, James Villarreal and Lui Wang of National Aeronautics and Space Administration – Lyndon B. Johnson Space Center / PT4 for their technical support and analysis help for this project.

References

1. Bayer, Steven E.: *SPLICER A Genetic Algorithm Tool For Search and Optimization User Manual*. A Product of the Software Technology Branch Lyndon B. Johnson Space Center, v.1,1991.

2. Bayer, Steven E.: *SPLICER A Genetic Algorithm Tool For Search and Optimization Reference Manual*. A Product of the Software Technology Branch Lyndon B. Johnson Space Center, v.1,1991.

3. Lea, Robert N.: *Automated Space Vehicle Control for Rendezvous Proximity Operations*. Telematics and Informatics, vol 5, no 3,1988.

4. Lea, Robert N., Jani, Yashvant , Hoblit, Jeff: *A Fuzzy Logic Based Spacecraft Controller for Six Degree of Freedom Control and performance Results*. Proceedings of AIAA '91, held at New Orleans.

5. Rawlins, Gregory J.E.: *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, Inc. 1991.

6. Edwards, H.C.; and Bailey, R.: *The Orbital Operations Simulator Users Guide*. LinCom Corporation, ref.LM85-1001-01, June 87.

7. Karr, Chuck: *Genetic Algorithms for Fuzzy Controllers*. AI Expert, March 1991.

Fuzzy Multiple Linear Regression -- A Computational Approach

by

C. H. Juang*, X. H. Huang, and J. W. Fleming
Department of Civil Engineering
Clemson University
Clemson, SC 29634-0911

Telephone number: (803) 656-3322
Fax number: (803) 656-2670

A paper prepared for presentation and publication
in
The NAFIPS '92
Puerto Vallarta, Mexico
December 15-17, 1992

*The presenter and to whom correspondence be addressed.

INTRODUCTION

Conventional regression analysis is a statistical tool for describing relationships between variables. If a large and representative data set is available, a "good" relation might be established using an appropriate model. If the statistical properties such as the coefficient of determination (R^2) meet certain criteria of "good" fitting, the relation obtained from the regression analysis may then be used for "making predictions." The regression technique is, indeed, a very useful tool for solving many engineering problems. However, there are situations where use of the conventional regression analysis is not feasible. For example, when data are imprecise, as is usually the case in many geotechnical engineering problems such as predicting the conductivity of clay liner, the conventional regression analysis is not applicable (Bardossy, et al., 1987, 1989). Another example concerns rules of thumb often used in engineering practice. These rules of thumb are, in loose sense, relationships between linguistic variables.

Fuzzy regression was perhaps first introduced by Tanaka et al (1982). Fuzzy regression analysis, as the name implies, uses the tools of fuzzy set theory to analyze fuzzy variables. Bardossy et al. (1987) extended fuzzy linear regression method by Tanaka et al. (1982) into nonlinear cases. In contrast to the statistical least-squares criterion, a fuzzy criterion based on a "vagueness" measure for the goodness of the regression was used in their approach. While this approach has been applied to solving many engineering problems, some questions remain to be answered. Among them are questions regarding uniqueness of the fitting, selection of the vagueness criteria, and the interpretation of fuzzy regression.

This paper presents a new computational approach for performing fuzzy regression. In contrast to Bárdossy's approach (1989), the new approach, while dealing with fuzzy variables, follows closely the conventional regression technique. In this approach, treatment of fuzzy input is more "computational" than "symbolic." The following sections first outline the formulation of the new approach, then detail the implementation and computational scheme, followed by examples to illustrate the new procedure.

EWR

FUZZY MULTIPLE LINEAR REGRESSION - A FRAMEWORK

Suppose that a set of limited number of observations, $(y, x_1, x_2, \ldots, x_n)$'s, is to be used to determine a relationship. If all variables are non-fuzzy, the conventional multiple linear regression involves fitting to the given data the following equation:

$$y = a_0 + a_1 x_1 + \ldots + a_n x_n \tag{1}$$

where $a_0, a_1, \ldots, a_n$ are the coefficients that minimize the sum of the squares of the residuals. These coefficients may be determined by solving the following equation:

$$\begin{bmatrix} n & \sum x_{1i} & \cdots & \sum x_{mi} \\ \sum x_{1i} & \sum x_{1i}^2 & \cdots & \sum x_{1i}x_{mi} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \sum x_{mi} & \sum x_{mi}x_{1i} & \cdots & \sum x_{mi}^2 \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_m \end{pmatrix} = \begin{pmatrix} \sum y_1 \\ \sum x_{1i}y_i \\ \cdot \\ \cdot \\ \cdot \\ \sum x_{mi}y_i \end{pmatrix} \tag{2}$$

The coefficient of determination (R^2), a handy measure of goodness-of-fit (but not an absolute indicator), is defined as follows:

$$R^2 = (S_t - S_r) / S_t \quad (3)$$

where

$$S_t = \Sigma [y_i - (\Sigma y_i)/n]^2 \quad (4)$$

and

$$S_r = \Sigma [y_i - (a_0 + a_1 x_1 + \ldots + a_n x_n)]^2 \quad (5)$$

In the above equations, all the summation is performed for *i* from 1 to n. Equations 1 through 5 define the conventional linear regression based on the least-squares criterion. These equations operate on non-fuzzy data. As such, interpretation of results of a regression analysis is straightforward.

Now, suppose all the given data are fuzzy numbers. In order to follow the above least-squares approach, new mathematical operations must be defined for processing these fuzzy numbers. Although fuzzy arithmetics (Kaufmann and Gupta, 1985) such as addition, subtraction, multiplication and division of fuzzy numbers along with many other operations have been introduced, the efforts required to directly implement the above regression analysis by fuzzy arithmetics would be overwhelming. It appears that a simpler approach is warranted.

In the present study, the JHE method (Juang, et al., 1991) is adopted to create a new procedure for performing regression analysis of fuzzy data. In the JHE method, fuzzy numbers are often characterized by *beta-M* membership function, f(z), defined below (after Juang, et al., 1992):

$$f(z) = C\,(z-b)^{\alpha}\,(d-z)^{\beta}, \quad (6)$$

where

$$C = \{\alpha^{\alpha}\,\beta^{\beta}\,[(d-b)/(\alpha+\beta)]^{\alpha+\beta}\}^{-1}, \quad (7)$$

$$\alpha = p^2(1-p)/q^2 - (1+p), \quad (8)$$

and

$$\beta = (\alpha+1)/p - (\alpha+2), \quad (9)$$

and where

$$p = (\mu-b)/(d-b), \quad (10)$$

and

$$q = \sigma/(d-b). \quad (11)$$

Notice that the parameters b, d, μ, and σ in the above equations are the minimum, maximum, mean, and standard deviation of the variable z. The parameters α and β are positive real numbers. The *beta-M* function is essentially a beta probability density function normalized with respect to its maximum functional value such that its maximum functional value at the mode is 1.0. It is a bounded function and satisfies the conditions for a fuzzy number (i.e., normal and convex fuzzy subset). The *beta-M* function can be symmetric, skewed to right, or skewed to left in shape, and is suitable for representing various engineering parameters with ambiguity.

The regression analysis involving equations 1 through 5 is basically a deterministic model. In a deterministic model, if the input is fuzzy numbers, the output will also be fuzzy numbers. For the problem at hand, the coefficients a_0, a_1, ..., a_n, and R^2 obtained from regression analysis will be fuzzy numbers. Thus, the predicted value, *y*, obtained from Eq. 1 for a given *x*-vector (x_1, x_2, ..., x_n) will be a fuzzy number. Since the data are imprecise, the "goodness-of-fit" may be measured by some "fuzzy indicator," perhaps in an analogous form of the coefficient of determination used in the conventional regression analysis.

In this study, the approach for performing regression analysis of fuzzy data is illustrated in Figure 1. Each fuzzy number input is first "de-fuzzified" before being processed by regression equations (Eqs. 1 through 5). The Monte Carlo simulation technique is used to *select* a non-fuzzy, random value for a fuzzy variable based on its membership function. Having de-fuzzified fuzzy numbers into non-fuzzy values, a set of coefficients including a_0, a_1, ..., a_n and R^2 can be obtained through the conventional regression analysis. After a large number of sets of the coefficients are obtained, fuzzy numbers representing these coefficients can be "re-constructed." Detailed procedure to implement this approach is presented below.

PROCEDURE FOR FUZZY MULTIPLE LINEAR REGRESSION

The proposed procedure for performing a fuzzy multiple linear regression is based on the JHE method. This procedure is detailed in five steps as follows:

Step 1. For each input fuzzy data (membership function), determine its cumulative function by integration. Determine also the maximum functional value of these cumulative functions in this step. Repeat this step for all input fuzzy variables.

Step 2. Begin the simulation by generating a uniform random number. Then normalize the generated random number with respect to the maximum functional value of the corresponding cumulative functions obtained in step #1, followed by equating the normalized random value to the cumulative function, a non-fuzzy value for each input membership function can be back-calculated. This step de-fuzzifies all input fuzzy data into non-fuzzy data.

Step 3. Perform the conventional multiple linear regression described in Eqs. 1 through 5. This step results in a set of coefficients, including a_0, a_1, ..., a_n, and R^2. This completes one iteration of the computation.

Step 4. Repeat Steps 2 and 3 a large number of times. The number of repetitions or simulations needed for a satisfactory result may be estimated by a trial-and-error procedure.

Step 5. Determine the minimum, maximum, mean, and standard deviation of each of the regression coefficients based on the values obtained from Steps 3 and 4. For each of these coefficients, the four parameters (b,d,μ,and σ) are used to define the *beta-M* membership function (Eqs. 6 through 11). This step results in a group of membership functions that define the wanted fuzzy numbers that represent the coefficient a_0, a_1, ..., a_n, and R^2.

INTERPRETATION OF FUZZY MULTIPLE LINEAR REGRESSION

Fuzzy multiple linear regression may be interpreted just as we would in the case of the conventional multiple linear regression. For a given vector of fuzzy numbers (x_1, x_2, ..., x_n) , the corresponding value of the dependent variable y can be predicted with Eq. 1. Although the predicted value will be a fuzzy number rather than a crisp number, the principle and the procedure are no difference from their well-established counterparts of the conventional regression analysis. Fuzzy output reflects the uncertainty mostly in the input in this case.

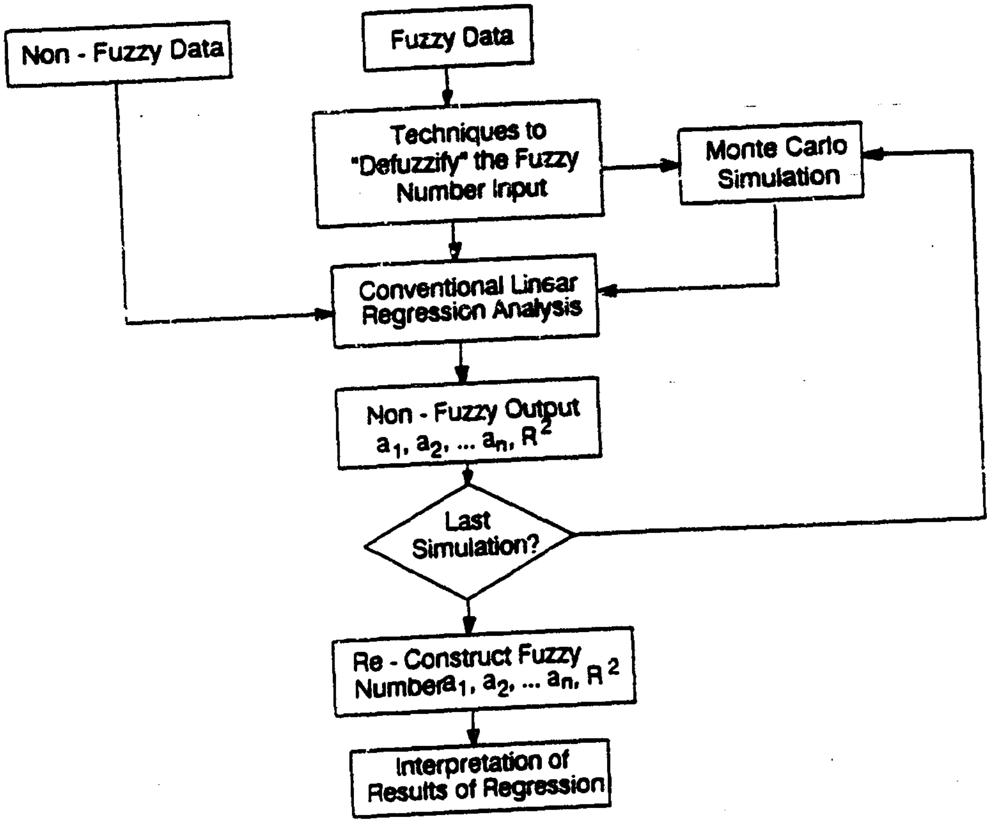

Non - Fuzzy Data
Fuzzy Data
Techniques to "Defuzzify" the Fuzzy Number Input
Monte Carlo Simulation
Conventional Linear Regression Analysis
Non - Fuzzy Output $a_1, a_2, \dots a_n, R^2$
Last Simulation?
Re - Construct Fuzzy Numbers $a_1, a_2, \dots a_n, R^2$
Interpretation of Results of Regression

Figure 1 A Schematic Diagram for the Proposed Approach for Fuzzy Regression

Interpretation of the "goodness-of-fit" is straightforward in principle. The coefficient of determination (R^2) in the conventional regression is renamed, tentatively here, as "fuzzy coefficient of determination" (FCD) in the fuzzy regression. The FCD is a fuzzy number describing the goodness-of-fit. This fuzzy number may be interpreted according to its maximum membership grade, by a mapping model which maps the resulting fuzzy number into a non-fuzzy value, or by translating the fuzzy number into a proper linguistic grade.

NUMERICAL EXAMPLES

Example 1 - This example is to perform a multiple linear regression of a set of non-fuzzy data as shown in Table 1. Here, the amount of water flows through a pipe per unit time, called discharge rate (Q), is assumed to be related to pipe diameter (D) and slope of the pipe (S) in a manner described by the following equation:

$$Q = a_0 D^{a1} S^{a2} \qquad (12)$$

Taking the logarithm of this equation yields

$$\log Q = \log a_0 + a_1 \log D + a_2 \log S \qquad (13)$$

Fitting this equation to the data shown in Table 1 yields the following results:

$$a_0 = 1.746,\ a_1 = 2.616,\ a_2 = 0.536,\ \text{and } R^2 = 0.999 \qquad (14)$$

The solution presented above was obtained using a computer program called FMLR (Fuzzy Multiple Linear Regression). The program FMLR implements the procedure and equations for performing fuzzy multiple linear regression presented earlier. When a non-fuzzy data set is input, the program functions like one which performs the conventional multiple linear regression. When the data is non-fuzzy, the relationship obtained from regression analysis is non-fuzzy, as reflected in this example. Equation 12 (with the coefficients determined through a regression analysis) is a form of Hazen-Williams equation commonly used in civil and mechanical engineering.

Table 1 Non-Fuzzy Data of Diameter, Slope, and Discharge Rate

D (ft)	S (ft/ft)	Q (ft^3/s)	D (ft)	S (ft/ft)	Q (ft^3/s)	D (ft)	S (ft/ft)	Q (ft^3/s)
1.0	0.001	1.4	1.0	0.01	4.7	1.0	0.05	11.1
2.0	0.001	8.3	2.0	0.01	28.9	2.0	0.05	69.0
3.0	0.001	24.2	3.0	0.01	84.0	3.0	0.05	200.0

Example 2 - The problem to be solved is the same as the one described in Example 1 except that the input data is fuzzy. The given data is shown in Table 2 where each datum is a fuzzy number. Each fuzzy number here is defined by four parameters b, d, α, and ß (Eq. 6). In addition, the mode m (the point at which the membership grade is 1.0) of each fuzzy number is shown. Note that an approximation of the fuzzy number used is a triangular fuzzy number defined by the parameters b, d, and m. Since the input data are fuzzy, a fuzzy regression analysis is performed. Results of the fuzzy regression analysis using FMLR are shown in Table 3. Each coefficient (a_0, a_1, a_2, or R^2) is a fuzzy number characterized by the four parameters (b, d, α, and β) of the *beta-M* function defined earlier. The mode of the *beta-M* function is also shown as a reference.

Table 2 Fuzzy Data of Diameter, Slope, and Discharge Rate - Given as logarithms

log D (ft)			log S (ft/ft)			log Q (ft^3/s)		
b	d	mode	b	d	mode	b	d	mode
-0.10	0.10	0.00	-3.30	-2.70	-3.00	0.132	0.161	0.146
0.27	0.33	0.30	-3.30	-2.70	-3.00	0.827	1.011	0.919
0.43	0.52	0.48	-3.30	-2.70	-3.00	1.245	1.522	1.384
-0.10	0.10	0.00	-2.20	-1.80	-2.00	0.605	0.739	0.672
0.27	0.33	0.30	-2.20	-1.80	-2.00	1.315	1.607	1.461
0.43	0.52	0.48	-2.20	-1.80	-2.00	1.732	2.116	1.924
-0.10	0.10	0.00	-1.43	-1.17	-1.30	0.941	1.149	1.045
0.27	0.33	0.30	-1.43	-1.17	-1.30	1.655	2.023	1.839
0.43	0.52	0.48	-1.43	-1.17	-1.30	2.070	2.530	2.300

Note: In this example, the parameters α and ß for all fuzzy numbers are set to be equal to 2.42. According to Juang et al (1992), in this case, these beta-M fuzzy numbers take the form of a π-curve, a bell-shape bounded function.

Table 3 Results of the Fuzzy Regression Analysis for Example 2

For the coefficient a_0:				
$b = 1.743$,	$m = 1.746$,	$d = 1.749$,	$\alpha = 1.27$,	$\beta = 1.20$
For the coefficient a_1:				
$b = 2.587$,	$m = 2.616$,	$d = 2.643$,	$\alpha = 1.26$,	$\beta = 1.22$
For the coefficient a_2:				
$b = 0.531$,	$m = 0.536$,	$d = 0.541$,	$\alpha = 1.26$,	$\beta = 1.22$
For the coefficient R^2 (FCD):				
$b = 0.999$,	$m = 1.000$,	$d = 1.000$,	$\alpha = 1.01$,	$\beta = 0.00$

Example 3

In many engineering problems, the basis for deriving a solution often is some rules of thumb provided by experts. For example, the possibility of meeting the EPA requirements for constructing a clay liner for the purpose of containing hazardous wastes is often assessed with a set of rules of thumb. Symbolically, each of these rules of thumb is expressed as follows:

IF X_1 is A_{1j} and X_2 is A_{2j} and X_3 is A_{3j}

THEN Y is B_j.

Here X_1, X_2, and X_3 are linguistic variables representing some factors that are thought to have an important influence on the possibility of meeting the EPA requirements, such as the *plasticity index*, *colloid percentage*, and *swelling potential* of the clay used. The values of these linguistic variables, A_{1j}, A_{2j}, A_{3j}, and B_j, are some descriptions commonly used in the assessment of clay liner. For example, a rule of thumb may state:

IF the plasticity index is *medium*, and the colloid percentage is *high*,
and the swelling potential is *low*,

THEN the possibility of meeting the EPA liner requirements is *very high*.

Now let's assume a group of rules of thumb on this subject are available, as listed in Table 4. These rules may be used to establish a predictive equation for assessing the possibility of meeting the EPA liner requirements. To begin with, all possible values of the linguistic variables used in the model need to be translated into fuzzy numbers. The linguistic terms and their corresponding fuzzy numbers used in this example are listed in Tables 5 and 6. With data given in Tables 4, 5, and 6, a fuzzy multiple linear regression can be performed. The results of this analysis are listed in Table 7.

INTERPRETATION OF RESULTS OF FUZZY REGRESSION

If the domain or range over which the FCD (a fuzzy number) is defined is small, the mode of this fuzzy number may be used to represent the FCD. On the other hand, if the FCD is quite fuzzy, an interpreting model is required. For example, the centroid of the membership function may be used to represent the fuzzy number.

One possible approach for interpreting the obtained FCD is to translate the resulting fuzzy number into a linguistic grade. A dictionary of linguistic grades for the goodness-of-fit, with their membership functions pre-defined (such as the one shown in Table 8), may be used to describe the "goodness of fit." This may be done by calculating and comparing "Euclidean distances" between the resulting FCD fuzzy number and the pre-defined fuzzy numbers of the linguistic terms. The Euclidean distance is a measure of "similarity" between fuzzy numbers. Thus, the most appropriate translation is the one with the smallest distance or highest degree of similarity. A simple model for the Euclidean distance is as follows (Zimmermann, 1987):

$$d_j = \sqrt{\sum [\mu_{CFD}(x) - \mu_j(x)]^2} \tag{15}$$

where d_j = distance between the FCD and the pre-defined fuzzy number j,
μ_{FCD} = membership function that defines the FCD, and
μ_j = membership function that defines the fuzzy number j.

Table 4 Rules of Thumb for Assessing the Possibility of Meeting the EPA Requirements

Plasticity index (PI)	Colloid Percentage (CP)	Swelling potential (SP)	Possibility of meeting the EPA requirements
high	high	high	low
high	high	medium	medium
high	high	low	high
high	medium	high	low
high	medium	medium	low
high	medium	low	medium
high	low	high	very low
high	low	medium	low
high	low	low	medium
medium	high	high	medium
medium	high	medium	medium
medium	high	low	very high
medium	medium	high	low
medium	medium	medium	medium
medium	medium	low	very high
medium	low	high	very low
medium	low	medium	low
medium	low	low	medium
low	high	high	low
low	high	medium	medium
low	high	low	high
low	medium	high	low
low	medium	medium	medium
low	medium	low	high
low	low	high	very low
low	low	medium	low
low	low	low	medium

In Example 2, the degree of fuzziness in the data is small. As a result, the fuzziness in the resulting coefficients is small, and interpretation of the "goodness-of-fit" is easy. In this case, R^2 is essentially equal to 1.0, and the fitting (regression) is rated as "excellent." In Example 3, the input data is fuzzier, and the resulting coefficients reflect this fact. The FCD fuzzy number is between the fuzzy numbers representing "very good" and "excellent" listed in Table 8. The Euclidean distance between the FCD and "very good" is larger than that between the FCD and "excellent." Thus, the fitting (regression) is rated as "excellent."

Another approach to interpret the "goodness-of-fit" is to plot the predicted versus observed values of the dependent variable. However, the predictions and observations, both as fuzzy numbers, needed to be first "de-fuzzified." In this case, the "center of gravity" approach may be used.

Table 5 Linguistic Terms and Their Corresponding Fuzzy Numbers - Independent Variables

Linguistic Term (PI)	Fuzzy number			Linguistic Term (CP)	Fuzzy number			Linguistic Term (SP)	Fuzzy number		
	b	d	mode		b	d	mode		b	d	mode
high	25	40	30	high	20	30	25	high	25	40	30
medium	10	30	20	medium	5	25	15	medium	10	30	20
low	0	15	10	low	0	10	5	low	0	15	10

Note: In this example, the parameters α and ß for all *beta-M* fuzzy numbers are set to be equal to 2.42. Other membership functions such as triangular or trapezoidal shape function, if desired, may be used.

Table 6 Linguistic Terms and Their Corresponding Fuzzy Numbers - Dependent Variable

Fuzzy number parameters	Linguistic Grade for "Possibility"				
	very low	low	medium	high	very high
b	0.00	0.00	0.25	0.50	0.75
d	0.25	0.50	0.75	1.00	1.00
mode	0.00	0.25	0.50	0.75	1.00

Note: In this example, the parameters α and ß for all *beta-M* fuzzy numbers are set to be equal to 2.42. Other membership functions such as triangular or trapezoidal shape function, if desired, may be used.

Once a satisfactory fuzzy relation is established through a regression analysis, it may be used to predict the value of the dependent variable for given values of the independent variables. For example, the equation obtained in Example 3 for predicting the possibility of meeting the EPA requirements is as follows:

$$P = a_0 + a_1 \text{ (PI)} + a_2 \text{ (CP)} + a_3 \text{ (SP)} \qquad (16)$$

where P = the possibility of meeting the EPA clay liner requirements,
PI = the plasticity index,
CP = the colloid percentage,

SP = the swelling potential, and
a_0, a_1, a_2, and a_3 = the coefficients defined in Table 7.

With this equation, the possibility of meeting the EPA liner requirements may be estimated for a given set of conditions regarding the plasticity index, colloid percentage, and swelling potential of the clay used. Since the values of the three independent variables PI, CP, and SP, and the coefficients a_0, a_1, a_2, and a_3 are all fuzzy numbers, the evaluation of this equation involves fuzzy computations. However, this can easily be done using the JHE method--simply replacing step 3 in the FMLR procedure presented earlier with *ordinary* addition and multiplication (Eq. 16). The result of such computation would yield a fuzzy number as the possibility of meeting the EPA requirements. The methods used for interpreting the FCD may be employed to interpret this resulting fuzzy number, and the possibility of meeting the EPA liner requirements is thus assessed.

Table 7 Results of the Fuzzy Regression Analysis for Example 3

For the coefficient a_0:				
b= 0.351,	m= 0.702,	d= 0.905,	α= 1.29,	ß= 0.75
For the coefficient a_1:				
b= -0.0028,	m= -0.0025,	d= -0.0019,	α= 0.41,	ß= 0.83
For the coefficient a_2:				
b= 0.010,	m= 0.014,	d= 0.014,	α= 0.77,	ß= 0.00
For the coefficient a_3:				
b= -0.020,	m= -0.020,	d= -0.016,	α= 0.00,	ß= 1.20
For the coefficient R^2 (FCD):				
b= 0.69,	m= 0.90,	d= 0.91,	α= 1.76,	ß= 0.13

Table 8 Fuzzy Numbers and Linguistic Grades for Describing Goodness-of-Fit

Fuzzy number parameter	Linguistic Grade for Goodness-of-Fitting				
	poor	fair	good	very good	excellent
b	0.00	0.00	0.25	0.50	0.75
d	0.25	0.50	0.75	1.00	1.00
mode	0.00	0.25	0.50	0.75	1.00

Note: Here the parameters α and ß for all *beta-M* fuzzy numbers are set to be equal to 2.42. According to Juang et al (1992), in this case, these beta-M fuzzy numbers take the form of a π-curve, a bell-shape bounded function.

DISCUSSIONS

It is observed that the modes of the membership functions of a_0, a_1, and a_2 obtained in Example 2 are practically identical to the coefficients obtained in Example 1 where standard regression was performed. As such, it might be speculated that a general relationship may be established between the range or dispersion in the membership functions of fuzzy variables D, S, and Q and the dispersion in the membership functions of the resulting coefficients, a_0, a_1, a_2, and R^2. However, a series of sensitivity analyses performed in this study (not shown here) seem to reject existence of such a general relationship.

The number of simulations required to reach a "steady" result is about 1000 for the examples studied. The maximum number of simulations tried was 10,000. The effect of degrees of fuzziness in the input data on the regression results was also studied. The results indicate that the fuzzier the input data are, the fuzzier the resulting coefficients would be. More study to verify these points is needed.

SUMMARY AND CONCLUSIONS

A new approach and procedure for performing fuzzy multiple linear regression is presented. The procedure is based on the JHE method for processing fuzzy information in the setting of multiple linear regression. By treating the conventional regression as a deterministic process (model), the JHE method can be applied to perform the regression analysis of fuzzy data. While input data may be fuzzy, as is often the case in many real-world applications, the new approach including the computation algorithms is precisely defined and is non-fuzzy. The new approach appears to be able to properly establish fuzzy relations from a given set of rules of thumb, based on limited study. More study is needed to further verify the proposed approach.

ACKNOWLEDGMENT

The study on which this paper is based was supported partially by the National Science Foundation under Grant No. MSS-9020252 to Clemson University. Dr. W. Bridges of Clemson University, a statistician, reviewed the manuscript and provided many constructive comments. Mr. P. Ghosh of Clemson University provided the rules of thumb on assessing the possibility of satisfying the EPA clay liner requirements listed in Table 4.

REFERENCES

1. Bardossy, A., Bogardi, I. and Kelly, W.E., "Fuzzy regression for resistivity-conductivity relationships," Proceedings, NAFIPS-87, Purdue Univ., W. Lafayette, IN., 1987, pp. 333-346.
2. Bardossy, A., Bogardi, I. and Kelly, W.E., "Geostatistics utilizing imprecise (fuzzy) information," Fuzzy Sets and Systems 31, 1989, pp. 311-328.
3. Tanaka, H., Uejima, S. and Asai, K., "Linear regression analysis with fuzzy model," IEEE Transactions on Systems, Man and Cybernetics, SMC-12, 1982, pp. 903-907.
5. Kaufmann, A. and Gupta, M.M., Introduction to Fuzzy Arithmetic - Theory and Applications, Van Nostrand Reinhold Co., New York, 1985.
6. Juang, C.H., Huang, X.H. and Elton, D.J., "Fuzzy information processing by the Monte Carlo simulation technique," Journal of Civil Engineering Systems, Vol. 8, No. 1, 1991, pp. 19-25.
7. Juang, C.H., Huang, X.H., and Elton, D.J., "Modeling and analysis of non-random uncertainties - fuzzy set approach," J. Numerical and Analytical Methods in Geomechanics, Vol. 16, 1992.
8. Zimmermann, H.J., Fuzzy Sets, Decision Making, and Expert Systems, Kluwer Academic Publishers, Boston, 1987.

519-63
161859

N93-29575

p. 10

Incorporation of Varying Types of Temporal Data in a Neural Network

M. E. Cohen*,D. L. Hudson#

*California State University, Fresno, CA 93740
#University of California, San Francisco, 2615 E. Clinton Avenue, Fresno, CA 93703

ABSTRACT

Most neural network models do not specifically deal with temporal data. Handling of these variables is complicated by the different uses to which temporal data are put, depending on the application. Even within the same application, temporal variables are often used in a number of different ways. In this paper, types of temporal data are discussed, along with their implications for approximate reasoning. Methods for integrating approximate temporal reasoning into existing neural network structures are presented. These methods are illustrated in a medical application for diagnosis of graft-versus-host disease which requires the use of several types of temporal data.

INTRODUCTION

Neural network modeling has received renewed attention in recent years [1]. Advances in both hardware and software have made the use of these systems for large-scale practical purposes feasible [2]. Neural network use is expanding rapidly in numerous domains [3-5]. Medicine has been a prime area of application of decision support systems based on neural networks for a number of reasons [6-8], including the difficulty of developing a traditional knowledge-based system for complex medical applications. A number of researchers have also investigated incorporation of fuzzy variables and techniques of approximate reasoning into neural network structures [9-14], including a number dealing with medical decision making [15,16]. Only recently has some attention been paid to the incorporation of temporal information in neural network models [17-19]. Temporal data have different interpretations depending on the application, thus general techniques cannot be successfully implemented without examining the ultimate usage of each of these variables. For example, the most straight-forward usage of temporal variables is in partial differential equations in which the time variable is clearly defined in mathematical terms and requires no further interpretation. However, only a few applications are well-understood enough to lend themselves to modeling through differential equations. For other less well-understood subjects, other approaches must be taken.

One of the strengths of the neural network approach is that their basic structure relies on the architecture of biological nervous systems, concentrating on the structure of the individual neuron, as well as the massively parallel nature of biological nervous systems [20]. Unfortunately, the processing of temporal information is only partially understood in a biological sense. The operation of short-term temporal influences can be explained by inhibitory and excitatory biochemical influences at the synapses, which account for the handling of conflicting signals within very short time intervals. However, the longer term handling of temporal information, including memory itself, is still a major area of cognitive research. Unfortunately, the current level of knowledge pertaining to this aspect of biological nervous systems cannot provide a clear model for handling temporal information.

In the next sections, different types of temporal data are examined, followed by the definition of structures which will allow the incorporation of these variables into an existing neural network structure previously developed by the authors, followed by a discussion of the use of fuzzy variables to represent both temporal and state variables.

TYPES OF TEMPORAL DATA

In traditional applications, temporal data have been handled in a number of ways, depending on the application. In well-defined models, partial differential equations can be used to represent temporal variables in the same way as state variables. Another valuable approach is the use of state-space diagrams, using transition functions to lead from one state to another. In the development of decision making algorithms for areas such as medicine, in general not enough information exists either to define a differential equation model or state-space diagrams. For applications such as this in which the majority of available information is contained in accumulated databases, neural networks offer a natural means for development of decision models. Toward this end, it is useful to analyze the manner in which temporal information is important to medical decision making, and it fact to other areas of decision making which rely on numerous findings which are utilized to differentiate among categories.

Temporal data can be divided into the following categories, depending on which aspects of the data are important:

1. Δ Data: The change in value from the previous recording (examples: blood pressure, cholesterol);

2. Normalized Δ Data: The change in value relative to the time interval (examples: weight gain or loss, hemoglobin level);

3. Duration Data: The duration of time for which the finding persisted (examples: chest pain, fatigue).

4. Sequence Data: A particular sequence of events (examples: fever occurring before rash occurring before generalized fatigue, noun occurring before verb occurring before adjective).

Each of these variable types requires special handling, each of which is discussed in the following section.

A. Δ Data, Normalized Δ Data, and Duration Data

These data types can be handled in a straight-forward manner, according to the following schemes. Let $n(t_i)$ be the value of the nth variable at time t_i, and let

$$\Delta n = n(t_i) - n(t_{i-1}). \tag{1}$$

$$\Delta t = (t_i - t_{i-1}) \tag{2}$$

Assign a new node in the neural network for Δ data such that

$$p_n = \Delta n. \tag{3}$$

The original network is then expanded by the number of nodes required to accommodate the items for which the change is important. For normalized Δ data follow the same procedure as before, except let

$$q_n = \Delta n/\Delta t. \tag{4}$$

Duration data can also be handled simply, by establishing

$$r_n = \Delta t_0 = (t_i - t_0) \tag{5}$$

For duration data, the important parameter is the length of persistence of a finding. Thus the Δt_0 in this case is the difference between the current time and the time t_0 when the finding originally occurred. It should also be noted that time measures (e.g. minutes, hours, days, months, years) should be normalized for each application.

B. Sequence Data

This is the most difficult problem in that a new variable cannot be created to deal with this entity. A major modification must be made to the neural network structure for accommodating this type of reasoning. These data are handled by embedding a procedure at each of the sequence nodes. To analyze for the presence of a sequence, let f_i, $i=1,...,k$ be the ith finding out of k and let t_i be the ith time interval. Define the square k x k matrix

$$S = [s_{ij}] \text{ where}$$

$$s_{ij} = \begin{array}{l} 1 \text{ if } f_i \text{ occurred at time } t_j \\ 0 \text{ otherwise.} \end{array} \tag{6}$$

For a proper sequence,

$$s_{ij} = 1 \text{ if } i = j$$

Thus tr [S] = k if the proper time sequence occurred, where tr [S] is the trace of the matrix S. The value of node u_n according is then determined by:

$$u_n = \begin{array}{l} 1 \ \text{ if tr [S] = k} \\ 0 \ \text{ otherwise} \end{array} \tag{7}$$

IMPLICATIONS FOR APPROXIMATE REASONING

The above constructs assume crisp input. The following modifications can be made to accommodate fuzzy input.

A. Δ Data, Normalized Δ Data, and Duration Data

For these data types, there are two parameters which may assume fuzzy rather than crisp values: the time dependent finding $n(t_i)$ and the time interval itself t_i. The $n(t_i)$'s can be of four types: binary, categoric, integer, or continuous. In fact, for these types of temporal data, the values themselves are not important, only the differences in the values. (If the value itself is important, it is incluaed as a separate node in the network.) Thus the generalization of the difference operation is required. The most straight-forward generalization appears to be extended subtraction for fuzzy sets defined in [21]. According to the algorithm established by Dubois and Prade, this operation can be applied to continuous variables, with a simpler, direct computation possible for the discrete case. If the data itself is binary or categoric, these variables can first be fuzzified, if appropriate. In the case of normalized data, the extended division, also discussed in [21], can be applied. It can be shown that if M and N are fuzzy numbers [21], then

$$M \ominus N = M \oplus (-N) \tag{8}$$

will also be a fuzzy number, where $M \oplus N$ is extended addition, and

$$M \oslash N = M \otimes (N^{-1}) \tag{9}$$

likewise is a fuzzy number, where $M \otimes N$ is extended multiplication.

B. Sequence Data

For the sequence data, whether or not a series of events occurred in a given order is a crisp result. However, the degree to which the sequence occurred in the correct order can be considered. Instead of setting node u_n as in equation (7), consider

$$u_n = \{tr\,[S]\}/k \tag{10}$$

The definition provides a degree to which the sequence occurred in the required order. For example, consider the k x k matrix

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots 0 \\ 0 & 0 & 1 & 0 & \dots 0 \\ 0 & 1 & 0 & 0 & \dots 0 \\ 0 & 0 & 1 & 1 & \dots 0 \\ \cdot & \cdot & & & \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ 0 & 0 & 0 & 0 & \dots 1 \end{bmatrix} \tag{11}$$

Then $u_n = (k-2)/k$, the degree to which the required sequence was met. Each row in this matrix represents a point in time, and each column represents a symptom. $s_{ij} = 1$ if at time i symptom j is present.

EXAMPLE

The method is illustrated on a problem for graft-versus-host disease (GVHD) taken from [22], and used as the basis for a recent workshop [23]. GVHD is a disorder which can occur after any kind of transplant operation, ranging from an organ transplant to tissue transplants, such as bone marrow. The disease exists in three forms: acute, chronic, and syngeneic. It is a complex disease in which changes in symptoms over time are extremely important for diagnostic purposes. The objective of the neural network decision aid is to determine if the disease exists in any of its three forms, or not at all.

Fig. 1 shows a neural network for this problem. Nodes n_1 through n_{k1} are standard nodes, p_1 through p_{k2} are Δ nodes, q_1 through q_{k3} are normalized Δ nodes, r_1 through r_{k4} are duration nodes, and u_1 through u_{k5} are sequence nodes. The following are examples of each type of node for GVHD:

n_{i_1}: presence of total body erythroderma (standard node)

n_{i_2}: thrombocytopenia (standard node)

p_j: change in number of B cells (Δ node)

q_k: sudden weight loss (normalized Δ node)

r_l: continued thrombocytopenia (duration node)

u_m: u_{m1}: pruritic maculopapular rash (sequence node)
u_{m2}: gastrointestinal abnormalities
u_{m3}: liver dysfunction
u_{m4}: bleeding

Note that thrombocytopenia is important both for its presence and for the length of time for which it has been present. In this application, all times will be considered to be in months or fractions of months, and are given as offsets from the initial visit, which is considered to be 0.

To illustrate, consider the following values for the above example:

n_{i_1} = 0.9 (degree of presence of total body erythroderma

n_{i_2} = 1.0 (degree of presence of thrombocytopenia)

$p_j = \Delta n_{i3} = n(t_{i3})\text{-}n(t_{i3}\text{-}1) = 300 - 170 = 130$ (assumes crisp values)
(change in number of B cells)

$q_k = \Delta n_{i4}/\Delta t_{i4} = (nt_{i4} - nt_{i4}\text{-}1)/(t_{i4} - t_{i4}\text{-}1)$ (assumes crisp values)
$= (140 - 130)/(4 - 2) = 5$ (weight change/month)

$r_l = t_{i5} - t_{i5_0} = 4 - 1 = 3$ (assumes crisp values)
(continued thrombocytopenia)

u_m:

$$U = \begin{array}{c|cccc|} & u_{m1} & u_{m2} & u_{m3} & u_{m4} \\ \hline t_{i_{6_1}} & 1 & 0 & 0 & 0 \\ t_{i_{6_2}} & 1 & 1 & 0 & 0 \\ t_{i_{6_3}} & 0 & 1 & 0 & 1 \\ t_{i_{6_4}} & 0 & 0 & 0 & 1 \end{array}$$

$$u_m = \mathrm{tr}\,[U]/k = 3/4 = 0.75$$

These values then become the values of the input nodes. Along with known classification values, the appropriate weighting factors are determined through the learning algorithm.

The network is trained on data of known classification to determine weighting factors for each of these nodes, both from the input layer to the intermediate layer, and from the input layer to the output layer. The result of the process is a differential diagnosis in which the degrees of presence of each form of the disease can be ranked.

FUZZY NEURAL NETWORKS

Another issue in the establishment of fuzzy neural networks is the role of linguistic quantifiers [24]. Considering the above example, the entry for q_k "sudden weight loss" refers to the normalized Δ data node. Although the linguistic variable "sudden" is not handled directly by the neural network learning algorithm, this concept is adequately represented by the amount of weight loss over a given time interval. The algorithm uses this information through the supervised learning process to assign an appropriate weight to this finding.

For the example in the previous section for Δ data, consider the B cell count. Due to inaccuracies in laboratory analyses, these results can be considered fuzzy numbers. If we assume each reading to be a fuzzy triangular number centered around the given values, with an experimental error of 5%, then the fuzzy values would be:

$$n(t_{i_3}) = (285{,}315)$$

$$n(t_{i_3}\text{-}1) = (161.5{,}178.5)$$

Then equation (8) can be applied. Similar results may be obtained for the other variables.

In order to handle fuzzy triangular numbers for standard nodes which do not represent Δ data, the algorithm for handling input interval data described in the next section can be applied.

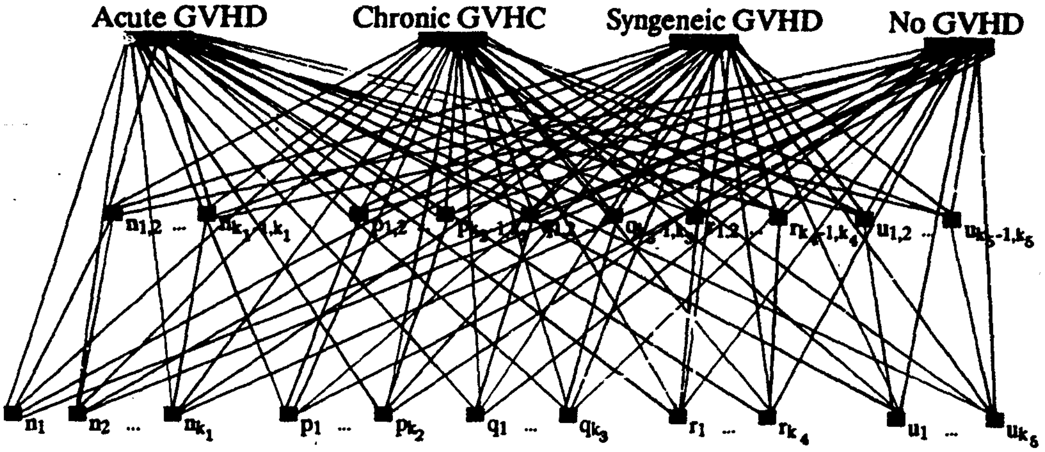

Acute GVHD
Chronic GVHC
Syngeneic GVHD
No GVHD
$n_{1,2}$ ... n_{k_1-1,k_1}
$p_{1,2}$ ... p_{k_2-1,k_2}
$q_{1,2}$ ... q_{k_3-1,k_3}
$r_{1,2}$ ... r_{k_4-1,k_4}
$u_{1,2}$ ... u_{k_5-1,k_5}
n_1 n_2 ... n_{k_1}
p_1 ... p_{k_2}
q_1 ... q_{k_3}
r_1 ... r_{k_4}
u_1 ... u_{k_5}

Figure 1: Neural Ne'work Structure Showing Temporal Data

NEURAL NETWORK MODEL AND LEARNING ALGORITHM

The heart of the neural network model is the learning algorithm. The topic of learning with fuzzy information has a long history, beginning with Wee and Fu's consideration of a fuzzy automaton in 1969 [25]. Kaufmann also considered fuzzy perceptrons in 1977 [26]. Zadeh suggested a different approach which used linguistically valued features [27]. Fuzzy isodata clustering algorithms have also been developed [28]. All of these approaches have relevance for neural network algorithms.

Following the learning algorithm previously developed by the authors, the temporal nodes are added, as shown in Fig. 1. If the node is fuzzy, an interval approach is taken, as previously described [29], in order to accommodate all extreme values. The algorithm permits the input of binary, categoric, integer, or continuous data, as long as an ordering exists for the categoric data. Variables which are not independent can also be handled directly. A summary of the interval data handling is given here.

Handling of Interval Data

In order to handle interval data as input, the following is proposed. For a data set with n variables, define a vector

$$x = [(x_1,y_1), (x_2,y_2),...,(x_n,y_n)] \qquad (12)$$

where (x_i,y_i) represents the interval range for the ith variable. The values for (x_i,y_i) will be determined by the input data in the training set for the learning algorithm. The objective is to obtain a decision surface which will separate data at any point in the interval. This can be accomplished if the extreme values are accommodated. In order to do, this all possible combinations of interval endpoints must be considered. For a data set with n variables, 2^n combinations will be produced. A new set of 2^n vectors is then defined:

$$z_k = [z_1, z_2, \ldots, z_n] \qquad k = 1, \ldots, 2^n \qquad (13)$$

where $z_i \in (x_i, y_i) \ni$ all possible combinations of x_i, y_j are generated for $i,j = 1, \ldots, n$. The learning algorithm is run for each of the 2^n cases. The weights attached to the decision surface which produces the poorest classification is chosen in order to form a robust model.

CONCLUSION

The neural network approach for development of decision support systems offers a number of advantages, including easy development of the knowledge base. As illustrated above, temporal data of several types can be accommodated into the existing framework. Variables can assume either crisp or fuzzy values. The resulting system can be used alone, or in conjunction with a knowledge-based expert system to bring to bear all relevant information whether from expert input or databases. In order to implement practical systems using interval data with large numbers of variables, it may be necessary to utilize parallel processing to establish models. The models themselves can be applied to new cases using standard sequential computers. Work is continuing in this area to streamline algorithms and to accommodate other types of fuzzy data into the system.

REFERENCES

[1] D.E. Rummelhart, J.L. McClelland, and the PDP Research Group, *Parallel Distributed Processing*, vols. 1 and 2, Cambridge, MA: MIT Press, 1986.

[2] G. Carpenter, S. Grossberg, The art of adaptive pattern recognition by a self-organizing network, *Computer*, 21, 3, 1988, 77-88.

[3] S.I. Gallant, Connectionist expert systems, *Communications ACM*, vol. 31, 2, pp. 152-169, 1983.

[4] B. Kosko, Hidden patterns in combined and adaptive knowledge networks, *Int. J. of Approximate Reasoning*, vol. 2, pp. 377-393, 1988.

[5] J.C. Bezdek, On the relationship between neural networks, pattern recognition and intelligence, *International Journal of Approximate Reasoning*, vol. 6, 2, pp. 107, 1992.

[6] R.C. Eberhart, R.W. Dobbins, Using neural networks in hybrid medical diagnostic systems, *Proc. EMBS, IEEE*, vol. 13, pp. 1470-1471, 1991.

[7] D. L. Hudson, M. E. Cohen, M. F. Anderson, Use of neural network techniques in a medical expert system, *Int. J. of Intelligent Systems*, 6,2, 213-223, 1991.

[8] C. L. D'Autrechy, J.A. Reggia, G.G. Sutton III, S.M. Goodall, M.A. Tagamets, "Developing connectionist models with MIRRORS/II," *SCAMC*, vol. 12, pp. 276-280, 1988.

[9] D.C. Kuncicky, A. Kandel, A fuzzy interpretation of neural networks, Proceedings, *International Fuzzy Set Association*, 3, 1989, 113-116.

[10] I. Hayashi, H. Nomura, N. Wakami, Artificial neural network driven fuzzy control and its application to the learning of inverted pendulum system, *Proceedings, International Fuzzy Set Association*, 3, 1989, 610-613.

[11] H. Takagi, I. Hayashi, Artificial neural network-driven fuzzy reasoning, *International Workshop on Fuzzy Systems Applications*, 1988, 217-218.

[12] R.R. Yager, Modeling and formulating fuzzy knowledge bases using neural networks, *Iona College Machine Intelligence Institute*, Report #MII-1111, 1-29, 1991.

[13] T. Saito, M. Mukaidono, A learning algorithm for max-min network and its application to solve fuzzy relation equations, *Proceedings, IFSA*, R. Lowen, M. Roubens, eds., 184-187, 1991.

[14] M. M. Gupta, M. B. Gorzalczany, Fuzzy neuro-computational technique and its application to modelling and control, *Proceedings, IFSA*, R. Lowen, M. Roubens, eds., 46- 49, 1991.

[15] E. Sanchez, Fuzzy logic neural networks in artificial intelligence and pattern recognition, *Int. Conf. on Stochastic and Neural Methods in Signal Processing, Image Processing and Computer Vision*, 1991.

[16] A.F. Rocha, M. Theoto, M. Theoto Rocha, Investigation medical linguistic variables, in *Proceedings, IFSA*, R. Lowen, M. Roubens, eds., 180-183, 1991.

[17] B. de Vries, J.C. Principe. A new neural network model for temporal processing, *Proc. EMBS, IEEE*, vol. 12, pp. 1439-1440, 1990.

[18] J.A. Villarreal, R.O. Shelton, A space-time neural network, *Int. J. of Approximate Reasoning*, vol. 6, 2, pp. 133-149, 1992.

[19] J.L. Elman, Finding structure in time, CRL Technical Report 8801, *Center for Research in Language, Univ. California, San Diego*, 1988.

[20] C. M. Butter, *Neuropsychology: The Study of Brain and Behavior*, Brooks/Cole, Belmont, CA 1968.

[21] D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, vol. 144, Academic Press, Orlando, 1980.

[22] J.L.M Ferrara. H.J. Deeg, Graft-versus-host disease, *N. Eng. J. Med.*, vol. 324, 10, pp. 667-674, 1991.

[23] D.L. Hudson, M.E. Cohen, Use of a hybrid expert system to diagnose graft-versus host disease, *in Artificial Intelligence in Medicine, AAAI Spring Symposium*, pp. 53-57, 1992.

[24] L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, 1, 3-28, 1978.

[25] W.G. Wee, K.S. Fu, A formulation of fuzzy automata and its application as a model of learning systems, IEEE Trans, Syst., Sci, Cybern, vol. 5. pp. 215-223, 1969.

[26] A. Kaufmann, *Introduction à la Theorie des Sous-Ensembles Flous, vol 4: Compléments et Nouvelles Applications*, Masson, Paris, 1977.

[27] L.A. Zadeh, Fuzzy sets and their application to pattern classification and cluster analysis, in *Classification and Clustering*, Academic Press, 1977.

[28] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *J. Cybern.*, vol. 3, 3, pp. 32-57, 1974.

[29] Cohen, M.E., Hudson, D.L., Approaches to the handling of fuzzy input data in neural networks, *IEEE Conference on Fuzzy Systems*, 93-100, 1992.

N93
29576
UNCLAS

/ 131860

N93-29576

FUZZY OPERATORS AND CYCLIC BEHAVIOR IN FORMAL NEURONAL NETWORKS.

Fuzziness may lead to chaotic dynamics.

E. Lábos[1], A.V.Holden[2], J.Laczkó[3], L. Orzó[1], A.S. Lábos[1]
[1]: Semmelweis Univ. Med. School, 1st Dept. of Anatomy, Neurobiology Unit of the Hung. Acad. of Sci., 1450 Budapest, Tűzoltó u. 58, HUNGARY; [2]: The Univ. of Leeds, Center for Nonlinear Studies, Leeds LS2 9JT, UK; [3]: Ludwig Maximilien Univ., Klinikum Grosshadern, Neurologische Klinik Munchen, GERMANY and Central Res. Inst. of Physics of Hung. Acad. Sci., HUNGARY

ABSTRACT - In formal neuronal networks (FNN) built of threshold gates, a unit step function is applied. It is regarded as a degenerated distribution function (DDF) and will be referred to here as a non-fuzzy threshold operator (nFTO). Special networks of this kind generating long cycles of states are modified by introduction of fuzzy threshold operators (FTO) i.e. non-degenerated distribution functions (nDDF). The cyclic behavior of the new nets is compared with the original ones. The interconnection matrix and threshold values are not modified. It is concluded that the original long cycles change: (1) fixed points, (2) shorter cycles or (3) as computer simulations demonstrate, aperiodic motion or chaotic behavior appears. The emergence of the above changes depend on the steepness of the threshold operators.

INTRODUCTION

A formal neuronal network (FNN) means now more than a McCulloch-Pitts network (1943): - (1) - the states of units and nets are fixed; - (2) - an interconnection matrix is fixed and synthesized through some process, here not by "learning"; - (3) thresholds are specified for each unit;- (4) a threshold function (a unit step function or a softer "S-shaped function" is finally applied. Thus the computation of the new network state is as follows:

$$s_{old} = s \longrightarrow sM \longrightarrow sM\text{-}\Theta \longrightarrow T(sM\text{-}\Theta) = s_{new} \qquad (1)$$

The sequence of s_i states can be generated by the iteration of this N network mapping which incorporates: (1) - interconnection matrix M;- (2) - threshold vector Θ ; and - (3) - threshold operator T. The networks may differ from each other by these objects. In learning processes an (M,Θ) sequence is generated in the hope of reaching a fixed network. In case of simulated annealing related to Boltzmann machines the steepness of a T - like function is changed to reach the limiting result.

In this study, the matrix M and threshold vector is fixed, the iteration is the only change. External input vectors which would transform the machine into a non-autonomous (open) system are not introduced. Learning (or adaptive synthesis) is not present. The network are however, very special. They were originally (Lábos, 1980-1987) synthesized in order to generate long finite cycles or to design networks of minimum number of required elements to a given length of cycles. The aim is to investigate the influence of the thresholds operator T to the length of cycles which appear during the iteration. For this reason "S-shaped" (still monotonic) operators are introduced as it is done in the "neurocomputer-science". This is the "fuzzy" aspect of the study.

1. MULTIPLE AND PRESENT MEANING OF FUZZY OPERATORS.

Among the possible meanings of "fuzzy" objects (sets, logic, grammar, languages, programs, environment, graphs, topology, etc.; see in Zadeh et al., 1975) only the following destination will be applied: as a membership function or degenerated distribution function. The non-fuzzy possibility is as follows:

$$\begin{aligned} u(r) &= 1 \text{ if } r \quad 0 \text{ and} \\ u(r) &= 0 \text{ if } r \quad 0 \end{aligned} \qquad (2)$$

Such a function is used in threshold logic or formal neurons and in their networks by coordinates. Since the 70-ies the concept was extended to "S-shaped functions" or the application of such non-idealized steps became a necessity. However, the consequences were not fully considered.

Lábos (1975) regarded the real nerve cells as special "measuring devices" included in their activity a special measuring procedure, generating a "measure space" (MS) in mathematical sense (Lábos, 1988; Halmos, 1974). The MS-s are closely related to distribution functions. Thus this generalization is plausible.

DEFINITION: - A real valued function F is called a distribution function if the following conditions are satisfied: - (1) F is increasing monotonically; - (2) - 0 F 1 ; - (3) F is semicontinuous from the left, i.e. lim F(r) = F(0) if r tends to zero from negative direction. The opposite continuity is not demanded, but permitted; - (4) - Optionally the differentiability of function F is also supposed.

REMARK: All distribution functions, including the discrete ones, belong to this category. Certain text-books demand one sided continuity from the opposite direction which does not make essential differences.

DEFINITION: A degenerated distribution function (DDF) is the function defined in eq. (2) and widely used in threshold logic. It is here called non-fuzzy threshold operator (nFTO). An arbitrary, non degenerated distribution function defining a Lebesgue-Stieltjes measuring space may have the name of fuzzy threshold operator.

REMARK: The "fuzzy" attribute makes the nomenclature applied in measure theory, logic or for membership-functions uniform. The semantical background is arbitrary. E.g. a response curve which may occur in a single natural or artificial neuron can be regarded either as a temporal average or it may represent a response of population of cells. A normalization to remain between O and 1 is useful, but can be omitted.

2.METHOD: COMPARISON OF BINARY THRESHOLD. GATE WITH FUZZY THRESHOLD GATES (FTO-s).

The FTO-s or non-DDF-s applied here are as follows:

$$T_1(x) = e^{kx}/(1+e^{kx}) = 1/(1 + e^{-kx}) \qquad (3)$$

$$T_2(x) = 0.5 + 0.5(e^{kx} - e^{-kx})/(e^{kx} + e^{-kx}) \qquad (4)$$

The binary threshold gate nets used here belong to a rather special class of networks (Lábos, 1980-1987). These are capable of generating transient-free behavior. In a more special moreover rarely occurring - i.e. non-generic - case the networks may generate long or even maximal cycle lengths. This means in an n-neuronal nets $L = 2^n$, the number of binary vectors in the state space. The nets were synthesized on the basis of a Theorem (Lábos, 1984, 1987) and were searched with computerized selection. Examples are presented in the quoted works from n = 1 to n = 9 dimensions (the number of neurons in a network). These finite nets in the actual autonomous case are non-chaotic. We will see however, that the behavior of these nets may become suspiciously chaotic as soon as FTO-s like T_1 and T_2 are introduced. In Fig 1. - 3. the so called code-trajectory of such networks is presented - as a reference - which is the diagram built of the consecutive states as decimally coded numbers based on the separate vectorial states of the net (e.g. code(011011) = 27 and n = 6. The diagram consists of the lines of (x,x)-(x,y) and (x,y)-(y,y) where x and y are successive state codes. This is simple of a method representation, similar to Poincare and Lamerey diagrams used in dynamics. In "chaotic" cases the next state plot of (x,y) pair of state is used only by coordinates.

3. BASIC OBSERVATIONS. (Figure 1-6).

The observations refer to the FTO cases, since the binary case is more explored and plays here the role of reference for the new behavior. The comparisons of the two situations are here the essential methodical and conceptual procedure. The computer simulations of which examples are given show that the "exponentially" long ($L = 2^{cn}$) or maximal ($L = 2^n$) cycle lengths of state flows change radically if FTO-s are applied instead of the unit-step-function. If the parameter k in functions (3) or (4) are suitable an originally long cycle of a coordinate-flow - after transient states - may become a fixed point, a pair of fixed states or four clusters or four points. The four clusters occur at higher values of k and corresponds to the (0,0), (0,1), (1,0), (1,1) quaternio of pairs of successive coordinates of state vectors. Their transitions (i.e. a next-coordinate plot) in the binary, non-fuzzy case are not so interesting since only the few(8) transitions may occur: (0,0)-(0,0), (0,0)-(0,1), (0,1)-(1,0), (0,1)-(1,1), (1,0)-(0,0), (1,0)-(0,1), (1,1)-(1,0), (1,1)-(1,1). In the fuzzy case (see all Figures) the state space becomes a continuum set and not only the coded vectorial flows shows interesting picture but also the coordinate-flows. For this reason and also because of hard representation, these phase-diagrams by components were displayed.

Bifurcation diagrams with the control parameter k of the steepness are also fabricated. The usual routes to chaos-like dynamics can be demonstrated. Such systems include n^2+n+1 numerical parameters because of the matrix, threshold vector and S-shaped operator where n is the dimension. E.g. at n=9, 91 different bifurcation diagrams are possible.

The study of non-monotonic operators instead of the distribution functions or threshold operators here is neglected since by changing the norm of matrix and threshold or value of k. Thus the working domain remains inside a bounded set.

The insight which can be gained from the various diagrams is a possibility of categorization of the diverse dynamical behaviors.

4. SHORTER CYCLE LENGTHS WITH FUZZY OPERATORS.

The most radical shortening of the cycles is the case when a maximal finite cycle becomes a fixed point. This occurs at very small absolute values of k. "Very small" seems to be different at different values of dimensions of the state vectors. Usually at higher dimensions smaller k-s still are capable of displaying complicated dynamical behavior.

5.OCCASIONAL LOSS OF PERIODICITY BY APPLICATION OF FUZZY OPERATORS.

As we cannot analytically prove that in such complicated dynamical systems which are presented here an aperiodicity in fact occurs, therefore the statement of the emergence of chaos is based on computer experience. This is a frequent situation in dynamics of chaos. The appearance of strange motion of coordinate flows is relatively simple at the investigated low dimensions. These attractors frequently consist of one, two, or a few disjoint or intersecting line-like plots. At certain regions the lines display thickenings (Birkhoff) which suggest complicated finer structure may be explored by zoom.

It is an undecided question whether in the cases which became chaotic(aperiodic), an infinite number of (un)stable periodic attractors depending on the initial value may occur or not. It is a question, what species of chaos occur (see e.g. in Holden, 1986; Kohda and Aihara, 1990, etc.).

6. OTHER EFFECTS.

Several phenomena were not detailed here: (1) Emergence of chaos from short cycle generating binary systems with or without transients; (2) How small the value of k could be , i.e. how "soft" or "elongated" S-threshold operator might coexist with the chaos; (3) It is to clarify the influence of the initial state to the attractor i.e. a study of the basin size of attractor. It is especially interesting if two or more loops or clusters occur in a next coordinate plot. It is often observed in such cases that the behavior is vibrating. Regular jumps occur between two cycles and therefore two cycles may have a unified basin of attraction.

7. ADVANTAGE OF FUZZY OPERATORS OR NOT?

The formal neuronal networks with the presented special class of matrices and thresholds are suitable for coding or for economical (small network) control of exponentially large number of effector organs. The advantage of fuzzy decisions for which this generalized truth functions can be used (see in Zadeh-Fu-Tanaka-Shimura, 1975) compared with the binary or many-valued logics is not yet completely explored. No doubt, the finite valued logical decisions can be played back to the binary case at least syntactically. The numerous values are justified if more than two meanings can be attributed to the variables.

However, in the case of continuous operators, the number of possible decisive cases becomes infinite or moreover continuum set. In technical implementation this can be a handicap or can be tolerated. Tolerance may be introduced by "digital (staircase-shaped)" decision operators dividing again the domains into sharply distinguishable subdomains.

The concept of fuzziness seems to be more applicable in contexts beyond those which were touched by this work (see in Zadeh, 1975; Bezdek and Sarkar, 1992).

No doubt, the present form of "soft threshold logic" as a continuous generalization of binary threshold logic and its relationship to the chaos appears to be a most promising theoretical subject. At the same time it might occur that chaos caused by the introduction of fuzziness or non-degenerated measuring operators may restrict the range of possible applications.

8. DISCUSSION

The main conclusion is that fuzziness - which is ubiquitously applied in neurocomputers - may introduce chaos (or even confusion; Mendes France, 1989) into the behavior of formal neuronal net or neurocomputer. After synthesis a tuning of operator is required to avoid chaos and implications. However, the transition between the sharply decisive dynamics of finite binary systems and chaos can be controlled by the slope parameter k. Chaos may occur in model networks (Labos, 1986; Derrida and Meir, 1988), the connection between the two paradigms merits attention. The message for neuromputer science is that it is not sufficient to synthesize a net let say by learning process, but still is necessary a tuning of the threshold operator. The real neural systems, display aperiodic but stable behavior. The presence of chaos in real nervous systems seems to be plausible (e.g. Freeman, 1987). But stability and reliability require deeper explanations.

ACKNOWLEDGEMENTS: This work was supported by OTKA 2621/1991 and ETT-T032/1990 Grants.

9. REFERENCES.

Bezek, J. C. and Sarkar, K.P. (1992): Fuzzy Models for Pattern Recognition. IEEE Press, Piscataway.

Carpenter, G. A.(1989): Neural Network Models for Pattern Recognition and Associative memory. Neural Network 2:243-257.

Derrida, B. and Meir, R. (1988):Chaotic behavior of layered neural network. Phys. Rev. A. 38(6): 3116-3119.

Freemann, R. (1987): Dynamical Models of Neural Function that Generate Chaos. IBRO-1987-Budapest, Neuroscience Suppl. to 22: 297S.

Grossberg, S. (1988): Neural networks and natural intelligence. Cambridge, MA., MIT Press.

Halmos, P.R. (1974): Measure Theory. Springer, Berlin.

Holden, A. (Ed): Chaos. Manchester UP, Manchester. 1986.

Kohda, T. and Aihara, K. (1990): Chaos in Discrete Systems and Diagnosis of Experimental Chaos. Trans. IEJCE E 73: 772-783.

Lábos, E. (1975): On the Significance of Wiring and Language in the Behavior and Information Processing of Neuronal Networks. In MTA Biol. Oszt. Kozl. 18: 325-340.

Lábos, E. (1980): Optimal Design of Neuronal Networks. In Neural Communication and Control. (Eds.: Székely, Gy. & al.) Adv. Physiol. Sci. 30: 127-153.

Lábos, E. (1984): Periodic and non-periodic motions in different classes of formal neuronal networks and chaotic spike generators. Cybernetics and System Research 2 (Ed.: R. Trappl), Elsevier. Amsterdam, pp. 237-243.

Lábos, E. (1987a): Chaos and Neural Networks. In: Chaos in Biological Systems (Eds.: Degn, H., Holden, A. V. and Olsen L. F.), NATO ASI Series A. Plenum Press. New York. pp. 195-206.

Lábos, E. (1987b): The most complicated networks of formal neurons. In Proceedings of IEEE 1st International Conference on Neural Networks. San Diego 1987, (Eds.: M. Caudill and Ch. Butler. Vol III, Network Architecture II. pp. 301-308.

Lábos, E. (1988): Asynchronous Versus Synchronous Computation in the Nervous System and Their Models. J. Molecular Electronics 67-77.

McCulloch, S. W. and Pitts, W. (1943): A logical calculus of the ideas immanent in the nervous system. Bull. Math. Biophys. 9: 127-147.

Mendes France, M (1989): Chaos implies confusion. In London Math. Soc. Lecture Note Series. Cambridge U.P., Cambridge.

Metropolis, N., Rosenbluth, A. W., Teller, M. N., and Teller, E. (1953): Equations of state calculations by fast computing machines. J. Chem. Physics, 21:1087-1091.

Simpson, P.K. (1990): Artificial Nervous Systems. Pergamon, London.

Zadeh, L.A., K-S, Fu, Tanaka, K. and Shimura, M. (1975); Fuzzy Sets and Their Application to Cognitive and Decision Processes. Academic Press., New York.

```
MATRIX:
 1 1 1-1-5 1-1
-5 1 1 1-1 1-1
-1 1 1 1-1-5-1
-1-5 1 1-1-1-1
-1-1 1 1-1-1 5
-1-1-5 1-1-1 1
-1-1-1-5-1-1 1
THRESHOLD:
-5-3-1-1-6-4 1
```

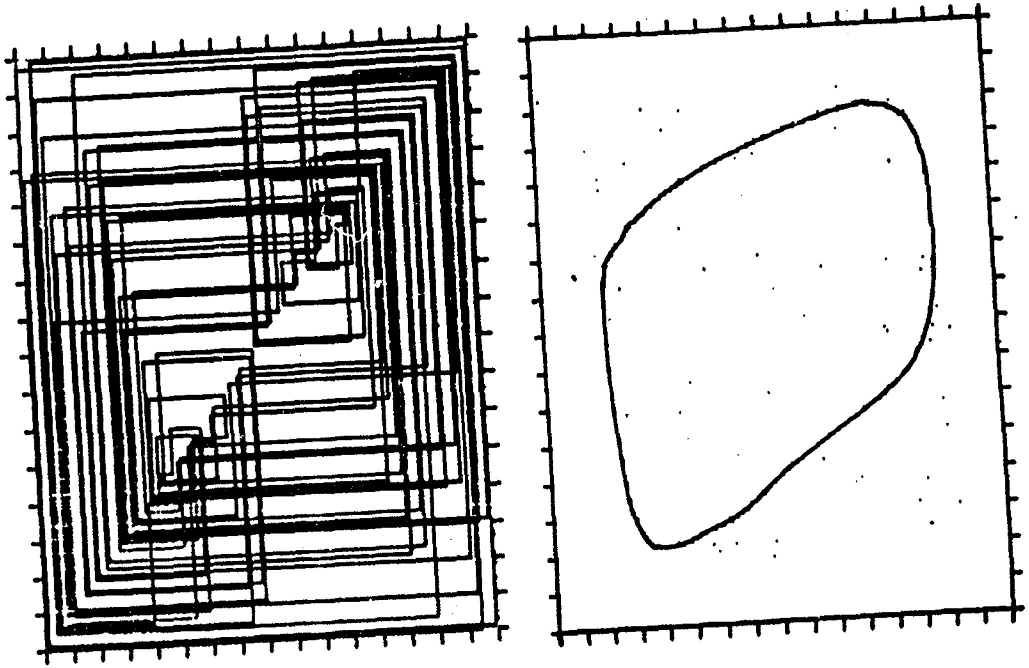
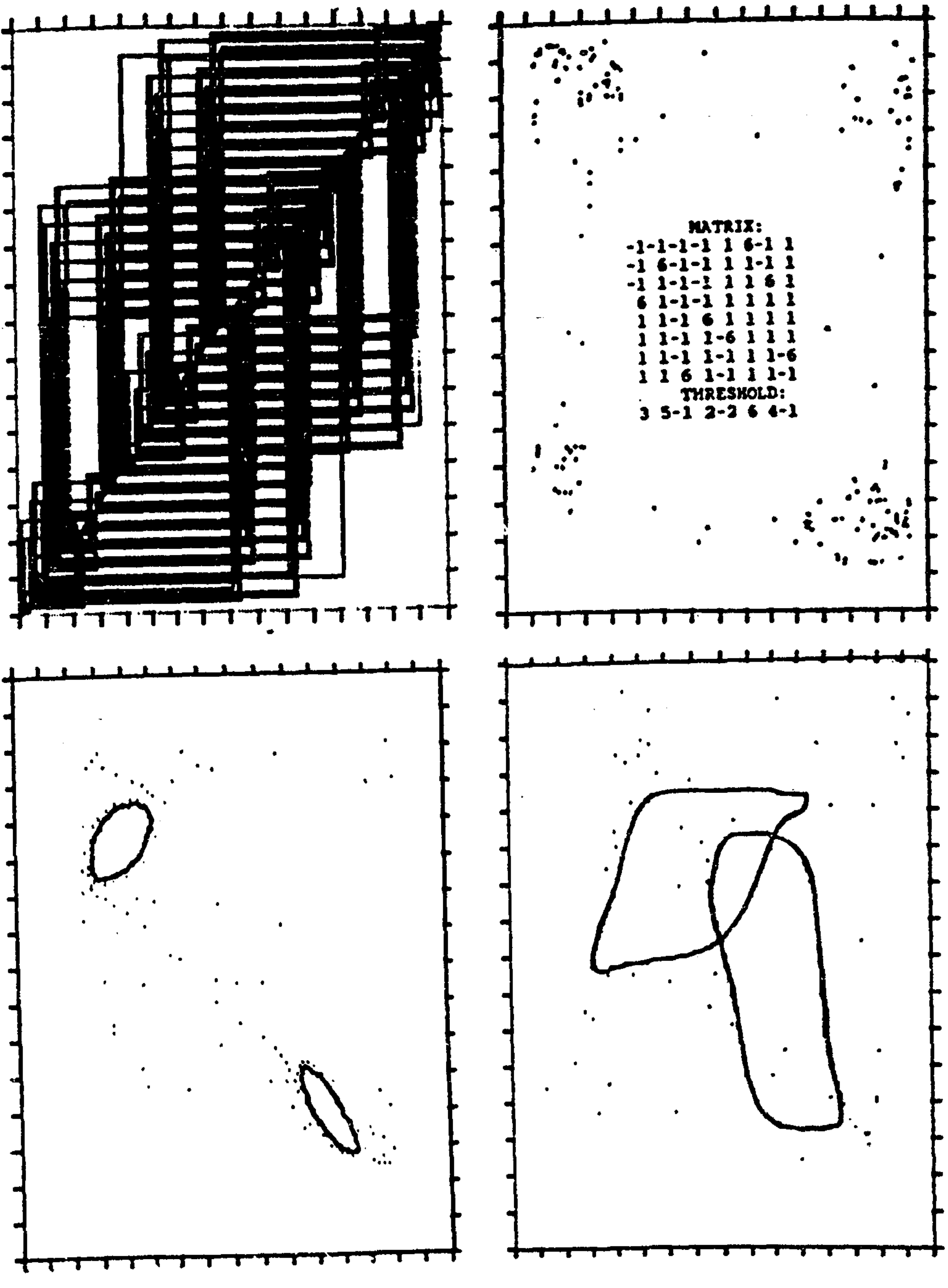


FIGURE 1 - Network of seven neurons. Next state plots. Left - (1) The matrix and threshold vector; Middle - (2) A code of vector sate trajectory of the cycle going through the zero vector, length is. L = 90; binary case; (3) Right: the flow of the first component is given with fuzzy operator T_1. The value of slope factor is k = 0.7.



```
MATRIX:
-1-1-1-1 1 6-1 1
-1 6-1-1 1 1-1 1
-1 1-1-1 1 1 6 1
 6 1-1-1 1 1 1 1
 1 1-1 6 1 1 1 1
 1 1-1 1-6 1 1 1
 1 1-1 1-1 1 1-6
 1 1 6 1-1 1 1-1
   THRESHOLD:
 3 5-1 2-2 6 4-1
```

FIGURE 2 - Network of eight units. Next state plots. (1) Upper left: The binary reference code trajectory of L = 256 length maximal cycle. (2) Upper right and later: Fuzzy operator of T_1 is applied. The matrix and threshold is inserted into the upper right frame, k = 0.865; component 1; (3) k = 0.72; component 4; (4) k = 0.8; component 1.

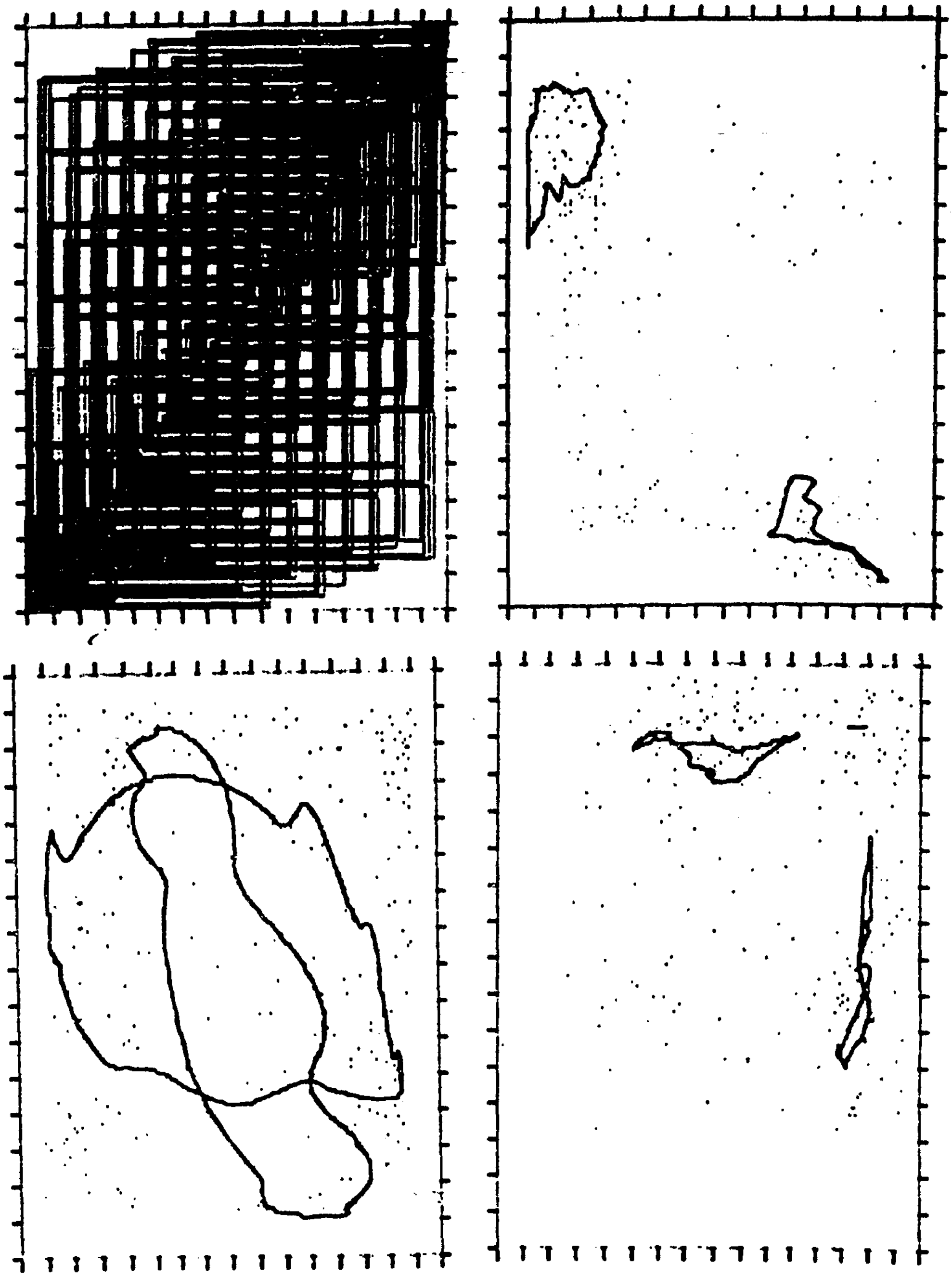
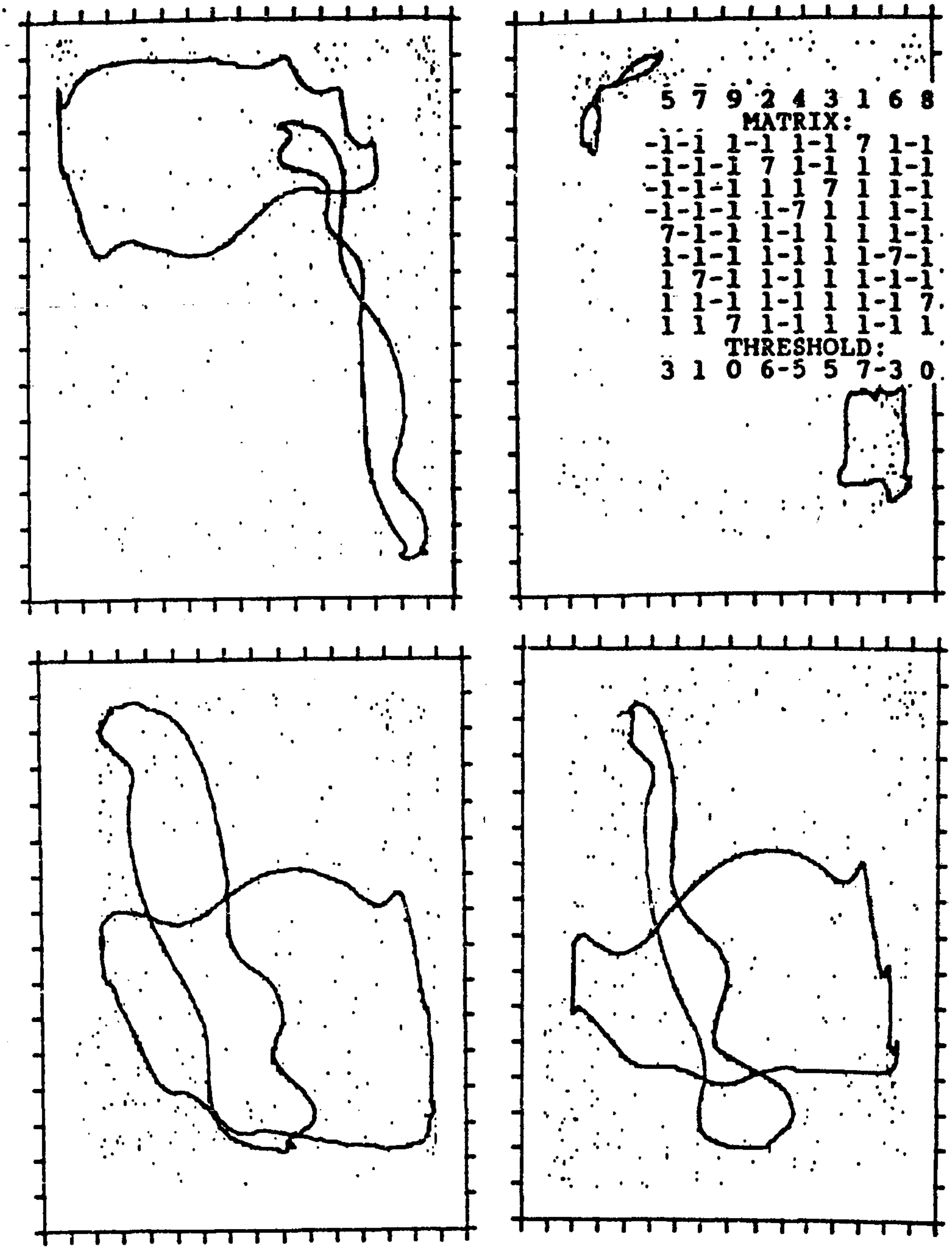


FIGURE 3 - Nine units. Next state plots. - For the non-fuzzy threshold operator: L = 512 (maximal). The matrix is in Figure 4. For fuzzy cases the components are successively as follows: 5th, 3rd, 4th.



```
 5 7 9 2 4 3 1 6 8
     MATRIX:
-1-1 1-1 1-1 7 1-1
-1-1-1 7 1-1 1 1-1
-1-1-1 1 1 7 1 1-1
-1-1-1 1-7 1 1 1-1
 7-1-1 1-1 1 1 1-1
 1-1-1 1-1 1 1-7-1
 1 7-1 1-1 1 1-1-1
 1 1-1 1-1 1 1-1 7
 1 1 7 1-1 1 1-1 1
    THRESHOLD:
 3 1 0 6-5 5 7-3 0
```

FIGURE 4 - FTO-next-state-plots. - Same as in Fig 3 but the flows of component 6th, 7th, 8th and 9th are displayed. In all cases the FTO is T_1 and k = 0.7.

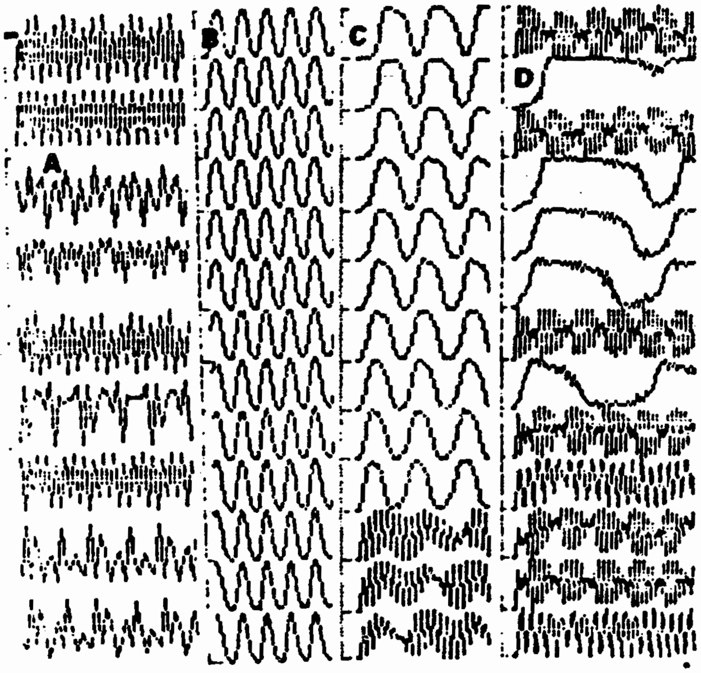


FIGURE 5 - Componentwise temporal diagrams: - (A) - n = 9, B-matrix, 579243168; 40; 0; k = 0.7. - (B) - n = 13; B-matrix, 123456789ABCD; 0; 0; k = 0.5. (C) - n = 13; B-matrix, 123456789ABCD; 7; 0; k = 0.5. - (D) - n = 13; B-matrix, 123456789ABCD; 5207); k = 0.5. See also Fig 6.

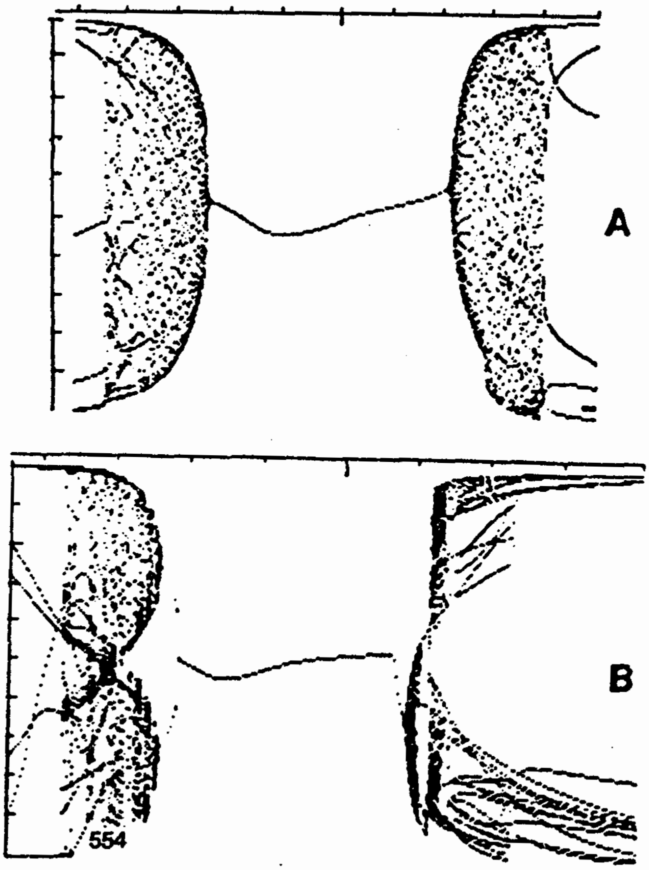


FIGURE 6 - Bifurcation diagrams. Control parameter is k. (A): n = 6; c = 2; Matrix A; cp = 351264; ng = 4; is = 0; (B): n = 8; c = 4; matrix A; cp = 42856137; ng = 6; is = 0. Non-fuzzy cycle length is maximal. Values on Y-axis are between 0 and 1; X- axis: goes is k and goes from -0.5 to +0.5. More detailed matrix specifications in Fig 5 and 6 see in the quoted publications of the first author.

NEURAL NETWORKS : A SIMULATION TECHNIQUE UNDER UNCERTAINTY CONDITIONS

M. Luisa Nicosia Mc Allister
Mathematics Department, Moravian College.
Bethlehem, PA 18018
Tel 215 8653187

Abstract

THIS PAPER PROPOSES A NEW DEFINITION OF FUZZY GRAPHS AND SHOWS HOW TRANSMISSION THROUGH A GRAPH WITH LINGUISTIC EXPRESSIONS AS LABELS PROVIDE AN EASY COMPUTATIONAL TOOL. THESE LABELS ARE REPRESENTED BY MODIFIED KAUFFMANN FUZZY NUMBERS.

§1 Introduction

Ever since F. Harary introduced the concept of implication digraph in his 1965 text, much of the theory developed has been of interest to applications involving transmission. In this era of knowledge engineering, artificial intelligence, and neural networks, the interest in graph theory has grown because it provides a source for problem solving techniques; see[15,18,21]. How do we include uncertainty in the representation and evaluation of transmission through a network? To answer this question, it is necessary to review some basic terms associated with well-known techniques used in the evaluation of the flow through a special type of graph. We want to emphasize that in using [13,14] the mathematics needed to incorporate uncertainty leads to easily applicable techniques which necessitate the discussion of fuzzy graphs. Thus we propose here to combine the principles of fuzzy set theory with those of graph theory. This combination may be applied to the problem of the evaluation of a transmission through a neural network. What imprecision do we have here that it is not handled by probability means? Because of the vagueness and uncertainty which occur in the simulation of realistically complex situations, we have to resort to techniques which can handle the vagueness of linguistic assessments. The modeling of neural networks is thus proposed by assuming that the concept of vertices being members of the vertex set and of arcs being members of the arc set is not crystal clear. It is susceptible to imprecision because of uncertain numerical evaluations,see[13], or because of linguistic assessments, see[14]. Thus using an approach according to fuzzy set theory in [12], it is possible to generate a simulation representing the imprecision, which is not of a probabilistic nature.

§2 Some basic terminology

The term graph in this context will be used to mean directed graph or digraph; see [1].

Definition 1: A graph is called stochastic if the following information is associated to each arc:

(1) The probability that this arc is selected;

(2) A random variable, such as time, is associated to each arc.

Definition 2: A stochastic graph is called a flowgraph if there exists a sink and a source,and if two boolean operators are associated to each vertex. The two operators are usually the AND or the ORELSE.

Why do we need these operators? These operators control the flow between vertices. The evaluation of the flow is one of the problems of interest when flowgraphs are implemented in a simulation procedure. Generally, in this type of applications, each arc entering a vertex represents an activity to be completed; see [2c]. When the AND operator is associated with that vertex, it means that no new activity can be pursued until all previous activities are completed. When the boolean operator is the ORELSE then only one of the entering activities must be completed before any new activity can be pursued. An additional requirement for a digraph to be called a flowgraph is that there ought to exist two special vertices. Recall that in graphs where more than one arc enters or leaves a vertex, we define for each vertex its indegree, the number of arcs entering that vertex, and its outdegree, the number of arcs leaving that vertex.

Definition 3: If there exists a vertex whose outdegree is equal to zero then it is called a sink. A vertex whose indegree equals to zero is called a source.

A flowgraph can concisely be defined as a stochastic digraph with two special vertices; a sink and a source. Flowgraphs have been successfully used to model the execution of activities as depicted by the digraph. It is then a natural extension to investigate their use when the network under investigation is a neural network. Surprisingly, no research efforts in such direction are known to this author. However, this is not the focus of this paper. As stated in the abstract, we propose here the use of fuzzy graphs as a technique to experiment with. The transmittance through a flowgraph were considered and solved by several authors, primarily using Mason's rule which is the best known; see [2]. A brief review of Mason's rule is given in the next section with some details. For additional details on algorithms and examples see[2,2a,2b,3].

§3. Path Transmittance in Flowgraphs: Mason's Rule

Let R be an n x n matrix where the value of each entry r_{ij} depends on a random variable. These values are obtained from the characteristic moment-generating function for the distribution of the random variable. Let P be a n x n probability matrix where p_{ij} equals the probability that the arc (i,j) is selected. As a brief summary, basically, the computation of the total transmittance along each path requires the search of all paths in the flowgraph from source to sink to be completed first. If we construct a new matrix, called the transmittance matrix, and denoted it by $T = (t_{ij})$ where each entry is the product of the a random variable with the corresponding probability which is associated to that arc; namely t_{ij} is the product of r_{ij} and p_{ij}. If we assume that the search of all paths from source to sink has already been made so that we know all the paths and that there are q paths from source to sink, then the total transmittance is computed according to Mason's Rule

$$\sum_{k=1}^{q} W_k \ (1 - \det(T_k)\} + (1 - \det(T)),$$

where T_k is a submatrix of T which is obtained from T by removing from it the row and the column that correspond to each vertex in the k-th path. The quantity W_k is the path transmittance of the k-th path. Let the fuzziness of each set be measured according to [4]; a method specifically designed for graphs.

§4. Fuzzy Graphs

The first to consider fuzzy graphs were A. Rosenfeld in [5,6,17],and R.T. Yeh with S. Y. Bang whose work is also included in reference [17]. Most authors, including this author in [2b,2d], defined a fuzzy graph as simply a graph whose adjacency matrix is replaced by the membership matrix $M = (m_{ij})$ under the convention that if the entry $m_{ij} \neq 0$, then the arc (i,j) has m_{ij} as the evaluation of the membership. There are two types of evaluations: the numerical, where we generally have $0 < m_{ij} < 1$, and that based on a functional interpretation, namely, $m_{ij} \neq 0$ is a mapping, or more specifically a fuzzy number.

Definition 4: A fuzzy number is a convex, normalized fuzzy set. At the conclusion of this work the suggested fuzzy number will be denoted z_n. We define a fuzzy graph as follows: Let V be the support set of the vertices. Let

$$m_V : V \in [0,1].$$

Then the fuzzy vertex set is denoted by

$$V^f = (V, m_V).$$

Similarly, let the fuzzy arc set be denoted by $A^f = (A, m_A)$ where m_A maps the support set A, which is the crip subset of the cartesian product of V with itself, into the interval [0,1]. A fuzzy graph is then the pair of fuzzy sets, written as, $G^f = (V^f, A^f)$. In all of the above and in the sequel, the superscript f is used to remind us that the set is assumed to be fuzzy. In this way, if $m_{ij} = 0$ then it means that there is no link between the vertex i and the vertex j. If the connection between the two links is not crystal clear then m_{ij} is either equal to some value between 0 and 1 or it is a fuzzy number.

A fuzzy graph was at first defined as a labelled graph, often without ever clarifying the necessary path algebra. A natural path algebra to use here is the MIN or the MAX operators which are commutative, associative, and distributive. For a full discussion on path algebras with examples, see [19;pp.85-88]. Note that the set of labels can consist of numerical values or of functional interpretations of linguistic expressions. In the former case, i.e. for numerical evaluations, the use of the MAX or MIN operators is fairly straightforward. If functional interpretations of linguistic expressions are used as label, then fuzzy numbers are used. However, in this case, the use of the MAX and MIN operators is not simple to use unless we devise a method for ranking fuzzy numbers.

A totally different definition of fuzzy graph is proposed here after a brief review of the homology of graphs.

§5 Background on the Homology of Graphs

Given a classical graph G = (V,A), we recognize two vector spaces. The first is the vertex vector space, V^f,and the second is the edges vector space A^f.

Definition5 A fuzzy Graph is a pair of vector spaces G^f = (V^f,A^f). Is this definition conflicting with the previous one? No, it simply identifies V^f and A^f for what they actually are: two vector spaces with the vector space operations defined by

$$(m+h)(v) = m(v) + h(v) \text{ and } (am)(v) = a\,m(v)$$

for every v in V and for any real or complex number a. The dimension of the vertex vector space V^f equals the cardinality of V. Note that in such case, the mapping m_V maps the set of vertices V into the set C of complex or real numbers and the mapping m_A maps the set of links A into C. Denote these vector spaces respectively by V^f and A^f. Thus a fuzzy graph is a pair of vector spaces, G^f = (V^f,A^f). Having stated that the use of the MIN and MAX operators is difficult when functional expressions are associated with each arc, it seems necessary to investigate how it is possible to solve the difficulty. Some authors have worked successfully on the ranking of fuzzy numbers; see [4] or the many papers by S. Ovchinnikov. Here, we propose the adoption of a somewhat easier solution because of a special type of fuzzy numbers we adopt. First we define the fuzzy numbers which will be used. They are called the Kauffmann integers.

§7 Some reasons for the term: Kauffman integers

First, they are called Kauffmann as they were introduced in [20]. Secondly, they are called integers because Kauffmann shows in his book that they form a Peano System. A brief review, for clarity of exposition, is given in §9. Note that this derivation is the same as the one given in [20]. The only new content given here consists of the propositions below and the fact that since the convexity requirement is satisfied but not the normality requirement. This is included in the definition of a fuzzy number. The functional representation is changed slightly.

§8 The Kauffman integers K_n

In this section a brief review is given mostly following Kauffman's derivation, first, a set denoted $(K^f)_\alpha$ is constructed for which we then derive some useful properties. They are called integers because Kauffmann also shows, see[20], that by defining a suitable operations $(K^f)_\alpha$ is essentially like the set of whole numbers N.

§9 The construction of the set $(K^f)_\alpha$

Let the elements of a set $(K^f)_\alpha$ be denoted by

$$(K^f)_\alpha = \{ K_1, K_2, K_3, \ldots\ldots, K_n, \ldots\ldots \}$$

where the subscript α is used because it is a parameter; any positive real number may be used. In this section, the elements K_n will be derived by defining a unary operation called the 'successor' operation. First we will define K_1 and then we will obtain K_n recursively from K_1 and K_{n-1} for $n \geq 2$. Then an explicit expression will be obtained for K_n.

(A) **Construction via Recursion**: Let the first element K_1 be defined according to the following: $f_1(x) = \alpha e^{-\alpha x}$, $\alpha > 0$ and let

$$K_1 = \{ (x , f_1(x))\}. \qquad (1)$$

Denote the next element by

$$K_2 = \{ (x , f_2(x)) : x \in [0,\infty) \},$$

where the function $f_2(x)$ is computed according to the following procedure:

$$f_2(x) = \int_0^x f_1(t) f_1(x-t)dt = \int_0^x \alpha e^{-\alpha t} e^{-\alpha(x-t)} dt = \alpha^2 \, x e^{-\alpha x} \qquad (2)$$

for $x \in [0,\infty)$. Any element of the set $(K^f)_\alpha$ is computed recursively according to the following:

$$f_n(x) = \int_0^x f_{(n-1)}(t) f_1(x-t)\,dt \qquad (3)$$

with

$$K_n = \{ (x, f_n(x)) : x \in [0,\infty)\}$$

for $n > 1$. Note that (3) essentially defines the desired recursion operation to derive the elements of $(K^f)_\alpha$.

(B) **Construction of an explicit Expression**:Both K_1 and K_2 have an explicit formula for $f_1(x)$ and $f_2(x)$. Can explicit formulae be found for the other elements, $K_n = (x, f_n(x))$? If so, we must be able to find an explicit expression for $f_n(x)$. Then, after a graphical interpretation, a few other facts will be established so that comparison operations can be easily identified and fairly simple to code.

Proposition 1: For all $n \geq 2$, we have

$$f_n(x) = \frac{\alpha^n x^{n-1} e^{-\alpha x}}{(n-1)!}.$$

Proof: The expression of $f_n(x)$ holds when $n = 1$ and $n = 2$. Assume that it holds for $(n-1)$ so that we have

$$f_{n-1}(x) = \frac{\alpha^{n-1} x^{n-2} e^{-\alpha x}}{(n-2)!}.$$

From the recursion definition it then follows that

$$f_n(x) = \int_0^x f_{(n-1)}(t) f_1(x-t)\,dt = \frac{\alpha^n}{(n-2)!} e^{-\alpha x} \int_0^x t^{n-2}\,dt.$$

Completing the evaluation of the integral we obtain precisely the statement of the proposition.

The advantage of the explicit form is that we can find the sketches of the membership function and its extrema and so might devise a method for ranking these numbers in the easiest possible way.

Proposition 2: For $n > 1$ each $f_n(x)$ has an absolute maximum M_n, with

$$M_n = \frac{\alpha\,(n-1)^{n-1}}{(n-1)!\, e^{n-1}},$$

and this occurs at $x = \frac{n-1}{\alpha}$.

Proof: Letting the derivative of $f_n(x)$ equal to zero gives the equation $n - 1 - a\,x = 0$ which has the desired value.

Since the second derivative of $f_n(x)$ is negative for this value of x, we have a maximum. It is an absolute maximum because $\lim_{n \to \infty} f_n(x) = 0$.

Note that this maximum needs not equal one.

Proposition 3: The sequence $\{M_n\}$, $n > 1$, is monotonically decreasing.

Proof: We have $M_2 = \frac{\alpha}{e}$ and $M_3 = \frac{\alpha}{e}\frac{2}{e}$. Therefore, $M_2 > M_3$. To show that the statement of the proposition is true, we must show that $M_k < M_{k-1}$ for all $k > 3$. Namely, we must show that

$$\frac{(k-1)^{k-1}\, e^{-(k-1)}}{(k-1)!} < \frac{(k-2)^{(k-2)}\, e^{-(k-2)}}{(k-2)!}.$$

To show that it is monotonically decreasing we have to show that the inequality $M_k > M_{k+1}$ holds for all k. Specifically we must show that the inequality

$$\frac{(k-1)^{k-1}\, e^{-(k-1)}}{(k-1)!} > \frac{(k)^{k}\, e^{-(k)}}{(k)!} \qquad (5)$$

is true by algebraically reducing it to a true statement. After simplification and taking the natural logarithm of each side, we have reduced the proof of the inequality (5) to the verification of the inequality

$$(k-1)\ln\frac{k-1}{k} + 1 > 0 \text{ or } (k-1)\ln\frac{k-1}{k} > -1 \qquad (6)$$

If (6) holds, (5) is true. Let $p(k) = k \ln \frac{k}{k+1}$. Since $p''(x) = \frac{k}{(k+2)^2} > 0$, then $p''(k) > 0$ because k is a positive integer. In addition, $\lim_{k \to \infty} p(k) = 0$ and it follows that $p(k) > -1$. Thus (6) is true.

To visualize the elements of $(K^{f})_{\alpha}$, namely some of the pairs (x, K_n), it suffices to sketch the functions $f_n(x)$ in the first quadrant since x, n, and α are all non-negative. The sketches in the figure at the end of this section provide a graphical interpretation of three elements: the fuzzy sets K_1, K_2, and K_3. In fact, the regions on the plane over the x-axis under the curves corresponding, respectively, to $f_1(x)$, $f_2(x)$ and $f_3(x)$.

§10 The Modified Kauffman Fuzzy Numbers Z_n

Note that there is no value of x for which f_n achieves the value 1 because there is no solution to $x = e^x$. Thus these numbers K_n are not normalized, and therefore are not fuzzy numbers either. Since convexity holds it is easy to verify that a minor modification of the membership function satisfies the normality condition.

In fact, let $g_n(x) = 1 + f_n(x)$,so that if $x = 0$, then $g_n(0) = 1$. Thus, the element $Z_n = (x, g_n(x))$, $x > 0$ are convex normalized fuzzy sets, and are therefore fuzzy numbers. The letter Z is used to remind us that Prof. L. A. Zade first introduced the concept of a fuzzy number and its computational application to fuzzy quantifiers in natural languages. See a detailed exposition in the text [4].

Can we define an ordering for the fuzzy numbers Z_n? This can be done, besides using its maxima values, also by other methods which are based on the next definition.

Definition 6: The height of the fuzzy set S^f on any interval $[a,b]$ of the real line is denoted by $h(S^f)$ and it is defined by $h(S^f) = \max\{f(x): x \in [a,b]\}$.

We can apply this definition to the modified Kauffman integers Z_n because of the propositions above. We not only know what their maximum value is but we also know where that maximum is. Since the maximum need not equal one, the use of the height is an alternative method which might be preferred over the use of the maximum. We can order these integers according to their height or according to the maximum M_n. We know what this maximum M_n by the previous propositions. Thus we have an easy computational method for their ranking.

§11 Some Concluding Remarks

It is a well-known fact that graph theory lends itself to applications. Often, we find computational techniques that are proposed without paying much attention to the coding, complexity, or storage difficulties. How do we store and manipulate graphs with so much information? If the coding language is Pascal, then it is recommended that each vertex is represented by a structure which contains information of the type: indegree, outdegree, etc. A similar structure is used for the set of links. All structures are linked to one another via linked lists.

A procedure called putnetwork outputs the graph and all information about it. The determination of paths is simplified because of recursion. The recursion is guaranteed to stop because there is a finite number of links; as branches are chosen, an array is passed down the recursion. Can we make use of fuzzy graphs? In previous work, we found applications for similarity relations which are important in building practical programs for fuzzy inferencing. We focus on what happens to the concept of similarity relations between distinct sets. The idea of similarity is no longer obvious. We find that k-partite graphs offer an alternative. An example of an application is included in previous work. Fuzzy bipartite graphs were a problem solving tool for J.Dockery and L. Mc Allister [2c,2d]. For example, in [2c], the authors focussed on what happens to the concept of similarity relations between distinct sets. The idea of similarity is no longer obvious. They find that fuzzy kpartite graphs offer a pictorial and computational alternative. An example of an application was included there.

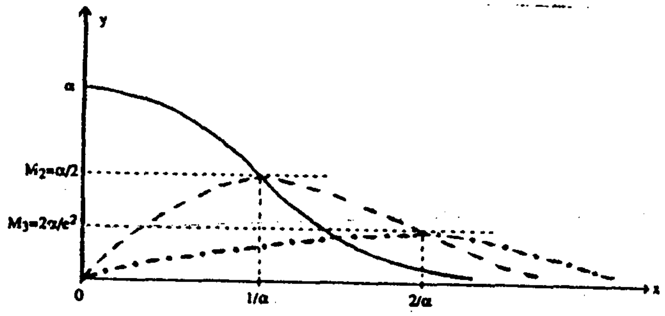


FIGURE : Sketch of the curves corresponding to γ_1 (——), γ_2 (- - - -), γ_3 (-·-·-)

P

REFERENCES

[1]-Harary Frank. R. Norman R., Cartwright D. "*Structural Models:An introduction to the theory of Directed Graphs*. Wiley ,1965.

[2]- Mc Allister. M.Luisa Nicosia. Cebulka Dickens Kathy Dickens S., & Galvin James"*Stochastic Networks":An Evaluation Technique and an Application". Proc. , Conf. on Modeling AND Simulation*. Univ of Pittsburgh. vol .13 , part 2. 1982. pp.775-778.

[2a] McAllister M.Luisa N., Cebulka K. D.S. "*The' Editorial Process of a BBimonthly : a Case Study*"" , Mc Allister M.Luisa Nicosia UMAP Journal. vol. 8. no. 4. pp.1-6. 1987.

[2b]- McAllister M.Luisa Nicosia "*A Measure of the Quality of Connection in a graph*. in Analysis of Information (J.C. Bezdek. Ed.), vol. 1. ch 9. CRC., 1984.

[2c] Mc Allister M.Luisa N. & Dockery J."*Similarity Relations* " Proceedings of the 1988 NAFIPS Workshop, San Francisco State University. San Francisco. CA., Ovchinnikov. ed.

[2d] McAllister Nicosia M.Luisa '*Fuzzy Intersection Graphs*" Comp.Math. Applic..vol. 15,no.10. 1988 pp. 871-886.1988.

[4] Dubois D. and PradeH. Fuzzy sets and Systems : Theory and their Applications. Academic Press.

[5]-Rosenfeld A. "*Fuzzy Digital Topology*" Info. and Control. 40. pp. .76-87.

[6]-Rosenfeld A. "The *fuzzy Geometry of Image subsets*". Univ. of Maryland Center for Automation Research, Comp .Sc, -TR1299, July 1983.

[7]- Shortliffe E. , Buchanan B. "A *model of inexact Reasoning in Medicine'* Math. Biosci .. 23. 1975, pp. 351-379.

[8]- Single K.& Gaynor M.& Halpern E. "*A n intelligent control strategy for computer consultation*" IEEE Trans. Pattern Analysis machine Intell. .PAMI-6., 2. 1984,pp. 129-136.

[10]- Weiss,Kulikowski C. & Amarel . "*A model based method for computer- aided decision making*", Artificial intelligence. 11. 1978,pp. 145-172

[11]-Xie W.X..& Bedrosian S."The *information in a fuzzy set and the relation* between *Shannon and fuzzy information*" , Proc. Conf. on Info. Sci. and Systems", John Hopkins Unversity. vol. 17. 1983. pp. 248-253.

[12]-Zadeh L.A. "*Fuzzy Sets*" Info. and Control. 8,1965, pp. 338-353.

[13]- Zadeh L.A. "The *role of Fuzzy Logic in the Management of Uncertainty* in *Expert Systems*" , Int. Journ. Fuzzy Sets and Systems" . 11.1983,pp.199-227.

[14]-Zadeh L. A."A *Computational Theory of Dispositions*" Proc. Int. Conf. on Computational Linguistics . Stanford Univ. ,1984. pp. 312-318.

[15] Tarjan R.E."*Data Structures and Network aAgorithms*" CBMS 44. SIAM 1983.

[16] Yager R." *Paths of Least Resistance in Possibilistic Production Systems*" Int. Journ. of Fuzzy Sets and System 28.2, 1986,pp. 121-132.

[17] "Fuzzy Sets and their Applications to Cognitive Decision Processes. (L.A. Zadeh. K.S. FU. ShimuraM. Tanaka. K. Eds.). Academic Press. NY. ,1975.

[18] Mc Allister M. N. & Dockery J., Jung P. "*Computing Similarity'* Proceedings of the 3rd IFSA Congress. Seattle. WA. July 1989.

[19] Carre' B. Graphs and Networks. Clarendon Press. Oxford. 1979.

[20] Kaufmann A. "Introduction to the Theory of Fuzzy Subsets. Academic Press. 1975.

[21] Wilf H.S. Algorithm and Complexity. Prentice Hall.1987.

N93
29578
UNCLAS

532-67
181832

p- 10

N93-29578

Incomplete Fuzzy Data Processing System Using Artificial Neural Network

Marek J. Patyra
Department of Computer Engineering
University of Minnesota
10 University Drive
Duluth, MN 55812-2496

Abstract

In this paper, the implementation of a fuzzy data processing system using an artificial neural network (ANN) is discussed. The binary representation of fuzzy data is assumed, where the universe of discourse is decartelized into n equal intervals. The value of membership function is represented by a binary number It is proposed that fuzzy data processing be performed in two basic stages. It is proposed that incomplete fuzzy data processing be performed in two stages. The first stage performs the "retrieval" of incomplete fuzzy data, and the second stage performs the desired operation on the retrieved data. The method of incomplete fuzzy data retrieval is proposed based on the linear approximation of missing values of the membership function. The ANN implementation of the proposed system is presented. The system was computationally verified and showed a relatively small total error.

1 Introduction

Fuzzy data processing systems that perform fuzzy operations can be implemented using standard or specialized software, but the ultimate way is to implement them in hardware. In fuzzy data processing systems, the major functions are performed by fuzzy processing elements like Min, Max, Bounded or Absolute Difference, etc., which can be connected in different ways (for instance, Min-Max-Min), depending on the desired structure. Building fuzzy data processing systems is attractive; however, in practice (i.e, in control systems) many incoming data to the system are incomplete (e.g., disturbed, noisy, or damaged). As a result, the output data generated by the system are wrong or contain an unacceptable errors that may cause a series of problems, especially in real critical applications.

Signal processing using a fuzzy approach has become more attractive during the last few years, when fuzzy sets and tools have been applied successfully to a variety of tasks. These tasks cover different areas of applications from speech and image processing to various pattern classifications [3]. Although the early stages of fuzzy signal processing mainly involving pattern recognition have been successfully developed, fuzzy methods for data processing (such as operations on various patterns) are yet to be developed. In the previous paper [2], ANN (Artificial Neural Network) realization of the fuzzy operations addition, subtraction, multiplication, division, minimum, and maximum, using neural networks, was studied. The conclusion of [2] indicates that the best results (in terms of average error) for fuzzy operations using ANN can be obtained when the operations are performed on nondegenerated fuzzy data. In contrast, the results of fuzzy operations using ANN performed on degenerated fuzzy data contain relatively high error. To overcome these disadvantages, the two-stage fuzzy data processing system is proposed in the present paper. The first stage performs the incomplete fuzzy data retrieval, while the second stage produces the results of a desired fuzzy operation.

The paper is organized in the following way. First the theoretical background for the retrieval of incomplete fuzzy data is given. Then the ANN realization of the retrieval stage is presented. The practical example, discussed in Section 2, shows the two-stage fuzzy data processing system (preprocessor performing fuzzy data retrieval and processor performing one of the discussed fuzzy operations). Finally, the simulation results for the system are presented, followed by the conclusions.

2 Fuzzy data retrieval

The discussion of the fuzzy data retrieval begins with the definition of the fuzzy number [1].

Definition 1. A fuzzy number $X=\{x_i\}$ is defined over a normalized set A on the real line R such that:

$$\exists x_i \in R, \sup \mu_A(x_i)=1 \qquad \text{(EQ 1)}$$

The $\mu_A(.)$ denotes a membership function of x_i in A and the x_0 referred to as the mean value of A if $\mu_A(x_0)=1$.
Assuming the discrete representation of the fuzzy number, the ordinary fuzzy number[1] can be described in the following way:

Definition 1A. Any fuzzy number X can be described in a finite domain $\{x_i\}$, by

$$X = \sum_{i=1}^{n} \frac{\mu_X(x_i)}{x_i} \qquad \text{(EQ 2)}$$

where i = 1,..., n and n defines the number of equal intervals into which the fuzzy number X is discretized and Σ denotes the union operation.

Based on the Definition 1A, there must be a mean value for the ordinary fuzzy number, and the EQ2 can be rewritten separately for the left and right intervals around x_0 as follows:

$$X = \sum_{i=1}^{k} \frac{\mu_A(x_i)}{x_i} + \frac{1}{x_0} + \sum_{i=k+2}^{n} \frac{\mu_A(x_i)}{x_i} \qquad \text{(EQ 3)}$$

(assuming $x_{k+1}=x_0$).Such a representation is called the "discrete representation" of fuzzy number. The special case of discrete representation, digital representation, is commonly used in most current applications of a fuzzy technology. Hence, universe of discourse is discretized into n intervals, each of which will be called "bit" by analogy with a digital representation of a number. However, any value from the interval [0,1] can be assigned to each bit of a digital fuzzy number. Additionally, it is assumed that the unimodal fuzzy numbers are discussed in this paper.

Definition 2. The degenerated fuzzy number Y' is the number with missing membership values μ_Y of one or more bit positions [2] (Fig. 1).

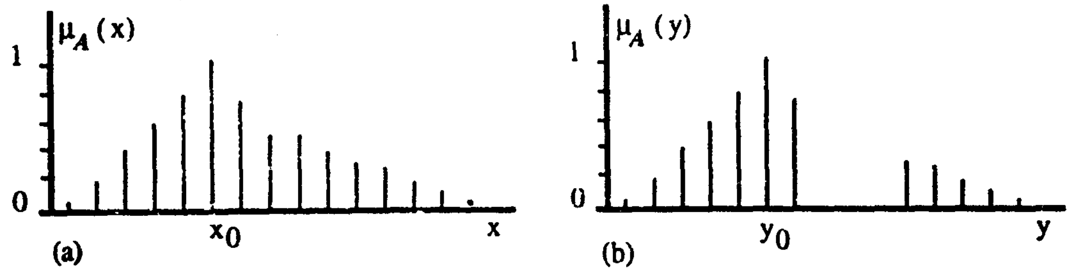


Figure 1. Digital representation of ordinary fuzzy number (a), and degenerated fuzzy number (b).

It is assumed in this paper that the discussed fuzzy numbers are unimodal. Let us now consider the degenerated fuzzy number Y' and their retrieval system.

[1] The term ordinary fuzzy number is used for fuzzy numbers in the sense of Definition 1.

[2] A special case of the degenerated fuzzy number with missing membership function values for all bits are not discussed in this paper.

Definition 3. The fuzzy number retrieval system (see also [6]) is defined by a triplet: (*Y'*, *Y"*, ρ), where *Y'* is a degenerated fuzzy number, *Y"* is a retrieved fuzzy number, and ρ is the retrieval function:

$$\rho\colon \mu_A(y') \rightarrow \mu_A(y'') : \forall\, y' \in Y' \text{ and } y'' \in Y'' \qquad \text{(EQ 4)}$$

where → represents a mapping relation.

Hereafter we use a simplified notation: $\mu_A(y) = \mu_y$ representing the membership function value of Y at the bit position i.

Definition 4. The fuzzy data retrieval function is defined by

$$\rho(\mu_{Y'}) = \mu_{Y''} \cong \mu_Y \qquad \text{(EQ 5)}$$

where *Y* is suppose to be an original fuzzy number which is free of missing bits (see Fig 1).

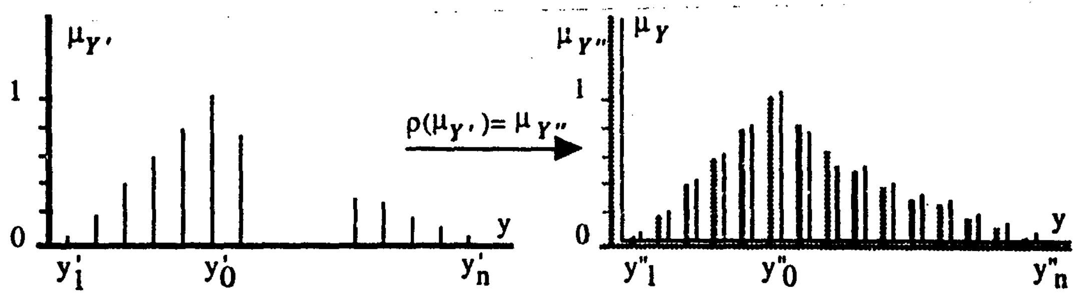


Figure 2. Interpretation of the definition of fuzzy retrieval system. *Y* represents the original number.

The characteristic of the retrieval function depends on the particular application. The implementation of a linear approximation technique, which seems to be good enough for most practical applications of fuzzy logic to the fuzzy data retrieval, is described below. In the process of approximation of the missing values for the membership function, two basic cases should be distinguished.

Case 1.

The membership function missing values correspond to the bits which are not: first $\{y_1\}$, last $\{y_n\}$ nor mean $\{y_0\}$. The number of bits with missing values in the left or right intervals can be arbitrary. In this case, the retrieval function simply extrapolates the membership function missing values based on the existing nearest values:

$$\rho(\mu_y) = y\frac{\mu_{y_s} - \mu_{y_{s+k}}}{y_s - y_{s+k}} + y_s \qquad \text{(EQ 6)}$$

Assuming that there are k missing bits, which start from *sth* bit, the membership function values can be obtained by incorporating $y_s, y_{s+1}, \ldots, y_{s+k}$ into EQ7. Note that, in such a case, it is necessary to approximate the membership function to the nearest available level of quantization. Let us consider the simple example where the number of bits with the missing membership function is equal to one (m=1). The bit number with the missing membership function in the degenerated fuzzy number is denoted by k. In such a case the membership function for the bit k can be approximated by[1]:

$$\mu_{y_k} = int\left(y_k\frac{\mu_{y_{k+1}} - \mu_{y_{k-1}}}{y_{k+1} - y_{k-1}} + y_{k-1}\right) \qquad \text{(EQ 7)}$$

[1] In this context the *int* function means the evaluation to the nearest quantization level.

If for any i ≠ k the $\mu_{y_i} \neq 1$, then the μ_{y_i} is set to 1 (see Fig.1.a).

Case 2.

The membership function missing values correspond to the bits which are: first $\{y_1\}$, last $\{y_n\}$ or mean $\{y_0\}$, or contain these bits. In this case, the retrieval function simply extrapolates the missing membership values.

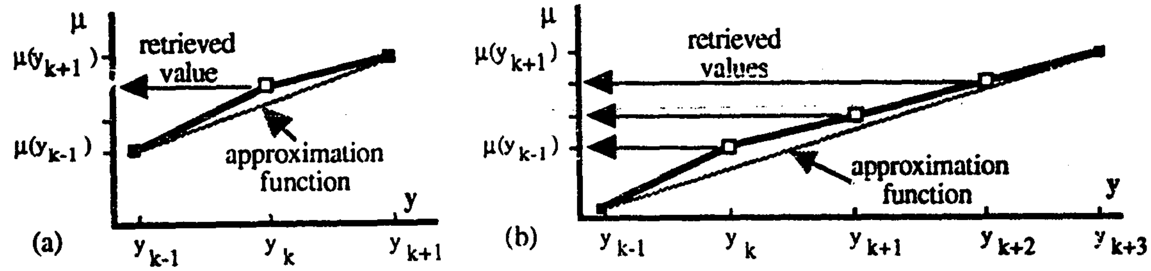


Figure 3. Example of membership function retrieval by linear approximation for a single bit (a), and for three bits missing (b). Black square represents known value and empty square represents retrieved value.

Let us consider the example where only single bits $\{y_1\}$, $\{y_n\}$, or $\{y_0\}$ are missing. In such a case the missing membership function values can be calculated using the formula given in EQ8. The only difference is that instead of calculating the membership function for the center bit, the one for the boundary bit is calculated.

The case where several bits have missing membership function values is not trivial and needs more discussion. It is proposed that the missing membership function values for the boundary bits (for left and right intervals) can be evaluated using the linear (prediction) function calculated based on the membership function values for the last two boundary bits[1]. Assume that there are two linear functions calculated for left and right intervals with the intersection point below 1 (see Fig.4 a). In such a case the mean value of a membership function is approximated to the nearest neighbor for any missing values of y_i and finally for the mean value $\mu(y_0)$ is set to 1: $\mu(y_0) = 1$.

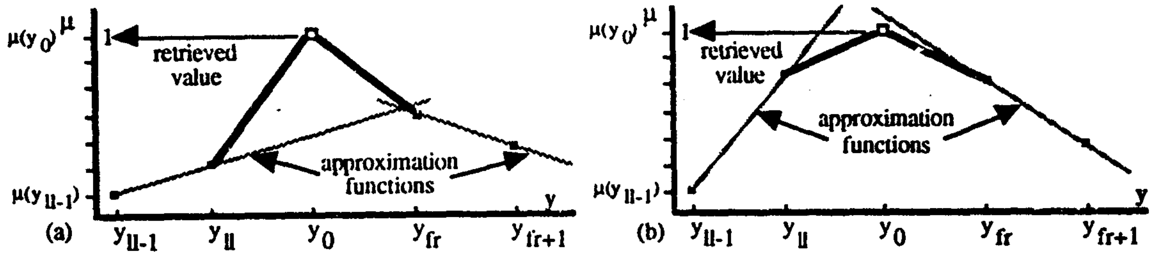


Figure 4. Example of membership function retrieval by linear approximation for several bits including the mean value: intersection point below 1 (a), intersection point above 1 (b).

Then new approximation functions are obtained based on the parameters of point (1, y_0), and last left [$\mu(y_{ll})$, y_{ll}], and first right [$\mu(y_{fr})$, y_{fr}] bits (see Fig.4 a). Assume that there are two linear functions calculated for left and right intervals with the intersection point above or equal to 1 (see Fig.4 b). In

[1] The procedure for determining missing values for membership function is discussed later in this sectio..

such a case the x coordinate for the mean value is approximate to the nearest y_i neighbor: then $y_i = y_0$ and $\mu(y_0)$ is set to 1: $\mu(y_0) = 1$. Then new approximation functions are obtained based on the parameters of point $(1, y_0)$, last left $[\mu(y_{ll}), y_{ll}]$, and first right $[\mu(y_{fr}), y_{fr}]$ bits (see Fig. 3 b). With these functions already calculated, the missing membership function can be evaluated using the formula giv in EQ7.

Case 2a.

The subcase 2a relates to the situation when the unimodal fuzzy number might have trapezoidai membership function. If such a case may origin when approximation functions (calculated based upon the existing values) "clamp" more then one quantization intervals (see Fig. 5). In this case the membership functions values can be simply approximate by setting their values to one.

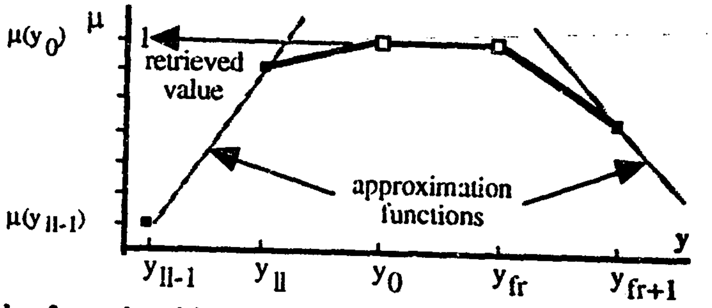


Figure 5. Example of membership function retrieval for the fuzzy number with trapezoidal membership function.

3 Linear approximation procedure and neural network

As it was proven in [5], any continuous function can be uniformly approximated by a continuous ANN using one hidden layer and with arbitrary continuous nondecreasing function. Such characteristic can be utilize to a task of the approximation of missing membership function values by the linear combination of existing values. Therefore, the linear approximation procedure should be used to obtain the training data set for ANN. Note that only selected membership function values are used for evaluation of missing values. The following architecture was assumed for the ANN implementation for fuzzy data retrieval. ANN consists of an input neuron layer (I), a hidden neuron layer (H) and an output neuron layer (O).

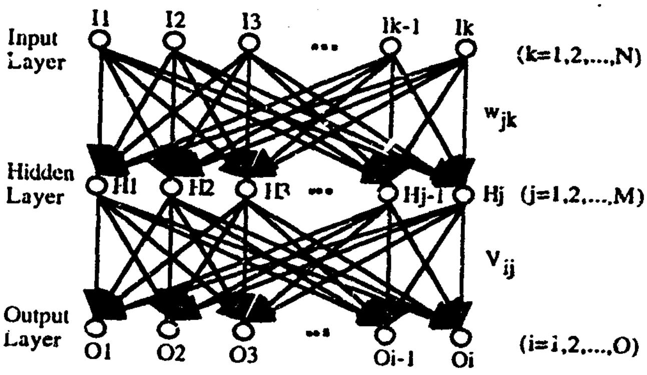


Figure 6. Structure of the ANN.

Each layer is completely connected, meaning that each neuron from a given layer is connected to all neurons of the next layer. A unique weight is associated with each connection. It was assumed that for fuzzy data retrieval the number of neurons in each layer equals to the number of bits in fuzzy number. Figure 6 illustrates the details of the discussed ANN configuration.

Now, the problem can be reformulated into whether it is possible to obtain sets of weights which minimize the total error of the linear combination of all existing membership function values with respect to the linear combination procedure. In the proposed method the output of the ANN which represents a retrieved membership function value μ_i^p for i-*th* bit can de described by:

$$\mu_i^p = g\left(\sum_{j=1}^{M} gV_{ij}\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\right) \qquad \text{(EQ 8)}$$

where g(.) denotes activation function, p denotes p-*th* input pattern, μ_k^p denotes the membership function value, V_{ij} represents weigh between *i-th* output layer element and *j-th* hidden layer element, while w_{jk} represents weigh between *j-th* hidden layer element and *k-th* input layer element (Fig. 6). The proposed linear approximation procedure used for i-*th* bit membership function value retrieval can be formally described as:

$$\mu_i^p = \sum_{j=i-1}^{i+1} (\alpha_j \mu_j^p) \qquad \text{(EQ 9)}$$

where α is an arbitrary coefficient (for i≠j). Then the typical error function can be given by:

$$2\varepsilon = \sum_{pi}\left[g\left(\sum_{j=1}^{M} gV_{ij}\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\right) - \sum_{j=i-1}^{i+1} (\alpha_j \mu_j^p)\right]^2 \qquad \text{(EQ 10)}$$

This is absolutely continuous, differentiable function of weights, so matrices **w** and **V** can be found minimizing the error using backpropagation method. For hidden-to-output and for input-to-hidden connections the steepest descent rule (which is a base for backpropagation method) gives:

$$\frac{\partial\varepsilon}{\partial V_{ij}} = \sum_{p}\left[g\left(\sum_{j=1}^{M} gV_{ij}\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\right) - \sum_{j=i-1}^{i+1} (\alpha_j \mu_j^p)\right] g'\left(\sum_{j=1}^{M} gV_{ij}\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\right) g\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right) \qquad \text{(EQ 11)}$$

$$\frac{\partial}{\partial w_{ij}}\varepsilon = \frac{\partial\varepsilon\ \partial V_j^p}{\partial V_j^p \partial w_{jk}} = \sum_{p} g'\left[g\left(\sum_{j=1}^{M} gV_{ij}\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\right) - \sum_{j=i-1}^{i+1} (\alpha_j \mu_j^p)\right] V_{ij}^p g'\left(\sum_{j=1}^{M} gV_{ij}\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\right) g'\left(\sum_{k=1}^{N} \mu_k^p w_{jk}\right)\mu_k^p \qquad \text{(EQ 12)}$$

Therefore, the system proposed before [2] was extended by the preprocessing stage incorporating ANN for degenerated fuzzy numbers retrieval followed by the Fuzzy Data Bus and the system for realization of fuzzy operation. The result also shows that it is not necessary to implement a special type of ANN, like that one suggested in [7], in order to obtain a good approximation of a fuzzy operation supplemented with the retrieval stage.

4 Computer simulation results

The data for training the retrieval ANN was prepared incorporating the linear approximation procedure in such a way that the degenerated fuzzy numbers were set to the input pattern and retrieved fuzzy numbers were set to the output pattern. Up to 70% of bits with missing membership functions was included in the set of 1024 degenerated fuzzy numbers (32 bits each). The PlaNet [4] simulator was used to train (with backpropagation procedure implemented for updating the weights) the ANN to the moment when average error (for all patterns) was less than 0.0001.

The fuzzy data processing system consists of two stages, the first stage performing fuzzy data retrieval (described in Section 3), and the second stage performing six basic fuzzy operations (described in [2]). The original architecture of the system described in [2] is illustrated in Fig. 7. In order to incorporate the retrieval stage, the original architecture was extended, by incorporating the additional stage.

The main goal of the application of this stage is to obtain retrieved fuzzy numbers available on the Fuzzy Data Bus for further processing. Note that the subsequent stages can perform simple (such as addition) as well as complex fuzzy operations such as inference, so such a system perfectly matches the general fuzzy modeling requirements. In the presented system the second stage networks are designed to perform addition, subtraction, multiplication, division, maximum and minimum (to be consistent with previously designed system). Table 1 summarizes results (average errors) obtained from the training pattern and testing pattern The training pattern contains randomly generated ordinary fuzzy numbers for six second-stage networks performing fuzzy operations. The testing pattern contains only degenerated, randomly generated fuzzy numbers, with missing membership values randomly distributed over the bit positions.

TABLE 1. Comparison of average errors for testing (degenerated) pattern obtained from second-stage ANNs performing: addition, subtraction, multiplication, division, maximum, and minimum.

Pattern	Addition	Subtraction	Multiplication	Division	Maximum	Minimum
Training	0.000353	0.000335	0.000395	0.000219	0.000382	0.000331
Testing (d)	0.004032	0.005630	0.009541	0.007680	0.006516	0.005766

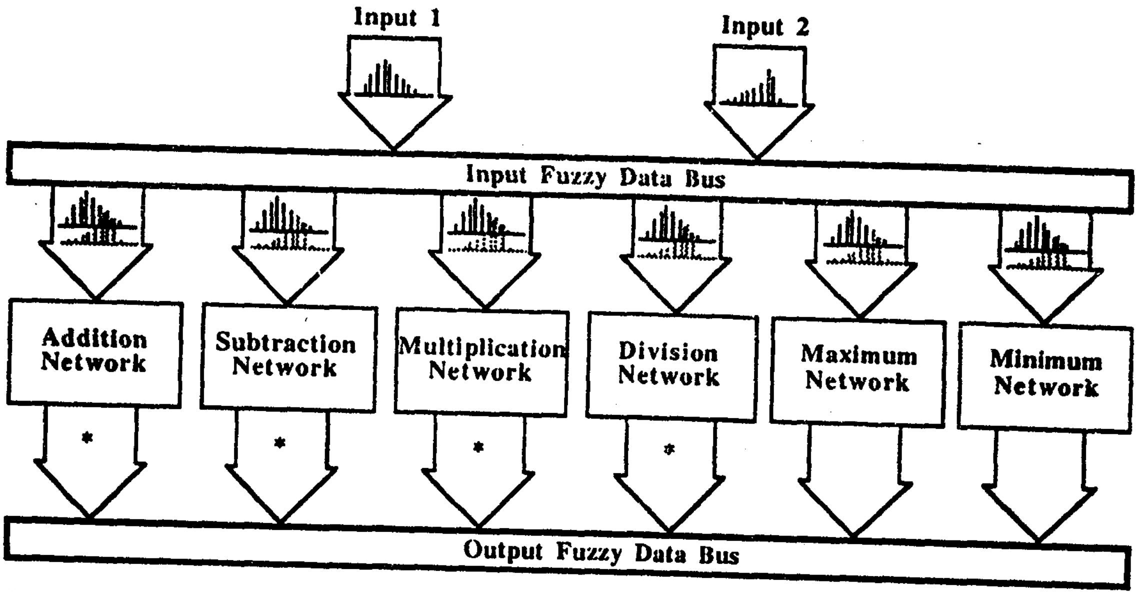


Figure 7. Architecture of the original fuzzy data processing system based on the ANN [2]. (* denotes the different data bus width as a result of fuzzy operations).

As one can see, the errors obtained from testing patterns including degenerated fuzzy numbers are five to ten times greater than original training errors. If we include in the simulation the ANN retrieval stage (average training error less than 0.0004), then the results obtained for testing patterns can be summarized in the Table 2.

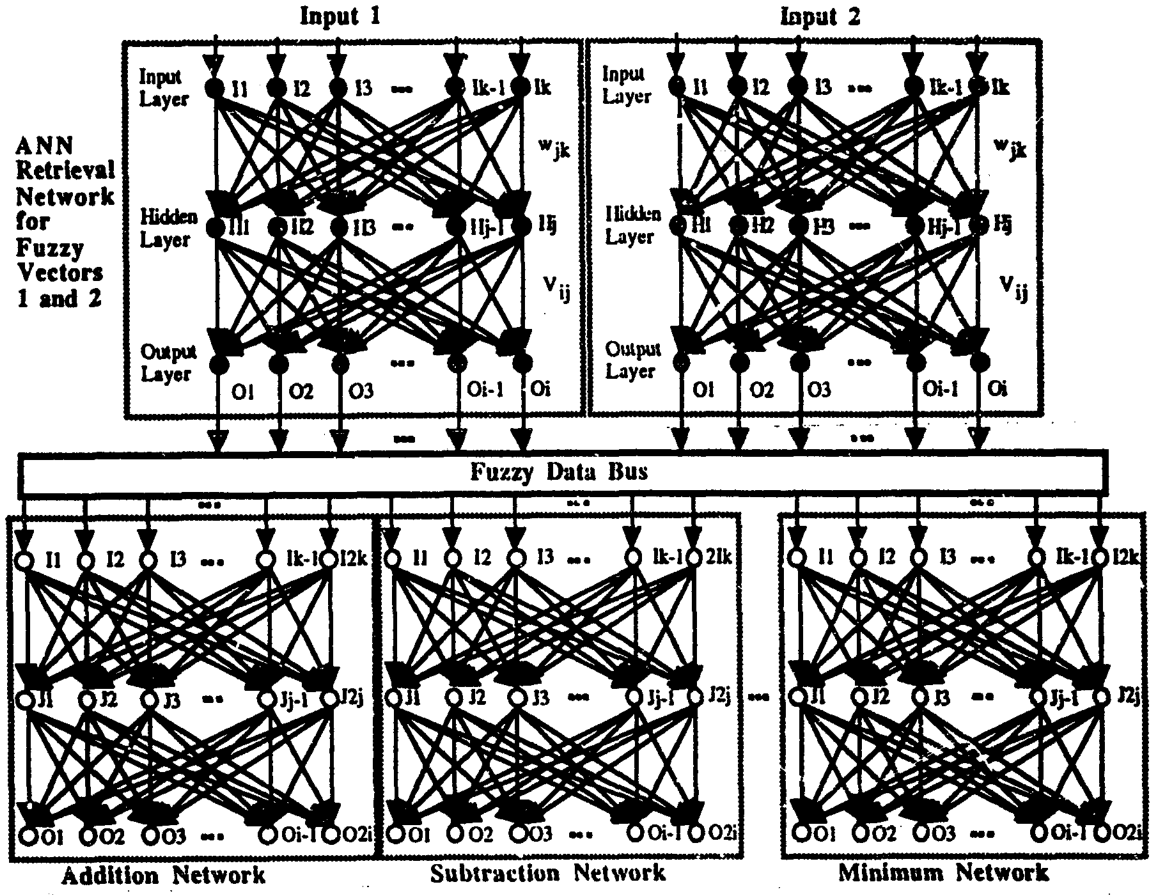


Figure 8. General architecture of fuzzy data processing system using ANNs.

TABLE 2. Comparison of average errors for testing (degenerated) pattern obtained from the proposed fuzzy data processing system including fuzzy data retrieval stage. Second-stage ANNs perform: addition, subtraction, multiplication, division, maximum, and minimum.

Pattern	Addition	Subtraction	Multiplication	Division	Maximum	Minimum
Testing	0.000643	0.000761	0.000577	0.000867	0.000742	0.000522

Figure 4 illustrates a fragment of the fuzzy data processing system, extracted from the original design (see Fig. 3), including the retrieval preprocessor. The values of membership function are coded in forms of sequence of squares. The area of a single square for a specific bit relates to the membership function value in such a way the largest square represents 1 and the smallest 0.1 (the empty place indicates 0). Two 32 bits long fuzzy numbers are set to Input1 and Input2 (data on Input1 is degenerated: missing membership function values for two bits). Then they are processed in the retrieval ANN (Hidden1(32b) and Hidden2(32b)). Finally, the retrieved numbers are displayed in Hidden3 layer (compare Input1&Input 2 and Hidden3(64b)). Then, these two ordinary numbers (first retrieved and second original) are processed in the subsequent network (Hidden4 (64b), Hidden5 (64b)), producing the result of operation (in this case, addition) at the Output (64b) [4].

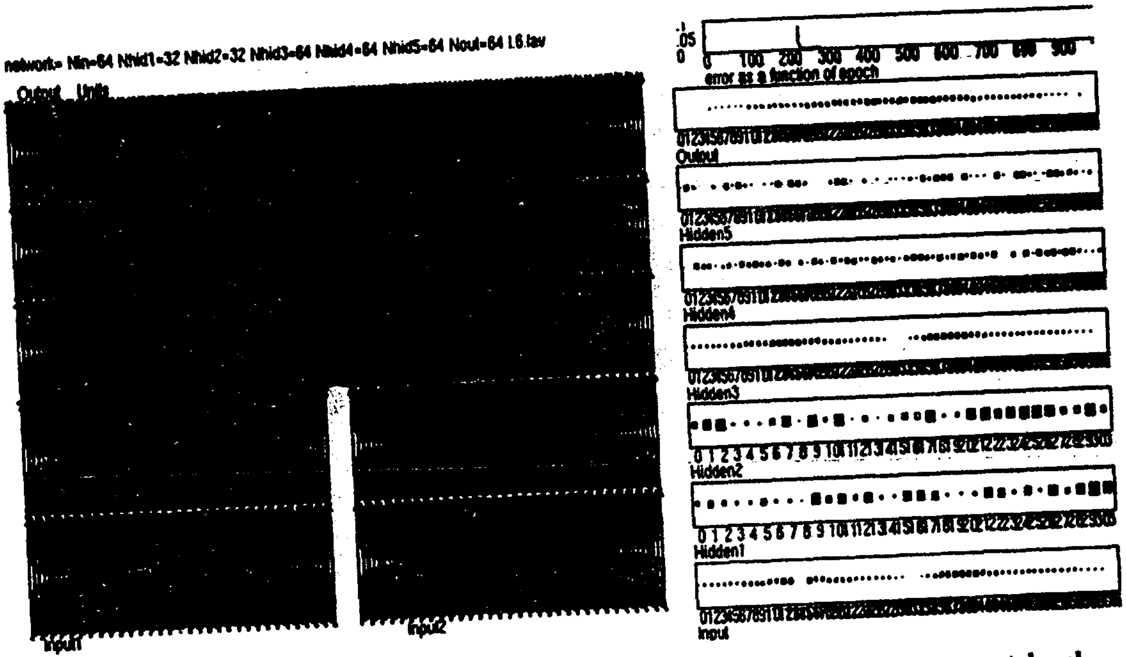


Figure 9. Example of the fuzzy data processing system applying fuzzy data retrieval preprocessor and the network performing fuzzy addition. (At the top, the trace of average error for training this part of a system is shown).

5 Conclusions

In this paper the implementation of a fuzzy data processing system using artificial neural networks is described. As it was verified in [2], the average errors for the testing patterns containing degenerated data were about two to five times greater than the average error for the normal testing data. In order to support fuzzy addition, fuzzy subtraction, fuzzy multiplication, fuzzy division, maximum and minimum for the degenerated fuzzy numbers the preprocessing stage devoted to fuzzy data retrieval was designed, trained, and incorporated into the fuzzy data processing system. The retrieval process was based on the linear approximation and prediction of the existing data for the incomplete fuzzy numbers. Such an architecture significantly improves the accuracy (up to ten times) of the results of operations performing on the degenerated fuzzy numbers; however, with the increase of missing membership function values, the average error also increases. The results of testing of the proposed system show that when up to 30% of membership function values are missing, the average error slightly increases (up to five times). Consequently, from 30% to 50%, the error is one order of magnitude higher. Finally, from 50% to 75%, the error can be ten to one hundred times greater than that for the training data. Due to these advantages the proposed architecture for fuzzy data processing systems may be very attractive in practical applications, especially in the case of processing heavily damaged fuzzy data by the real-time fuzzy logic controllers.

One should also stated the main advantage of ANN application to the real-time control consisting of the adaptive changing of the fuzzy model implemented through ANN, along with changing of the process under control. Such a feature is necessary when fuzzy controller is suppose to control dynamic, time-variant system.

6 References

[1] A. Kaufmann and M. Gupta, *Introduction to Fuzzy Arithmetic. Theory and Applications*, Van Nostrand Reinhold Company, New York, 1985.

[2] M. J. Patyra, "Implementation of Fuzzy Operations with Neural Networks", *IEEE Conference "Fuzzy and Neural Systems and Vehicle Applications'91"*, Tokyo, Japan, November 1991.

[3] H. Takagi and I. Hayashi, "NN Driven Fuzzy Reasoning", *International Journal of Approximate Reasoning*, pp. 191-212 (8), 1991.

[4] Y. Miyata, "PlaNet User's Guide", Computer Science Department, University of Colorado, Boulder, CO, 1991.

[5] G. Cybenko, Approximation by Superposition of a Sigmoid Function, *Mathematics of Control, Signals and Systems*, pp. 303-314, (2), 1989.

[6] S. Miyamoto, "Fuzzy Sets in Information Retrieval and Cluster Analysis" Kluwer Academic Press, Dordrecht, Netherlands, 1990.

[7] S. Horikawa et al., "A Study on Fuzzy Modeling Using Fuzzy Neural Networks", Proc. of International Fuzzy Engineering Conference, IFES '91, pp. 562-573, Yokohama, Japan, November 1991,

N93
29579
UNCLAS

S23-63
161563
p-6

N93-29579

Stochastic architecture for Hopfield neural nets

Sandy PAVEL
Polytechnical Institute of Iasi,
Bd. Copou, nr. 11, Iasi, ROMANIA

Abstract

An expandable stochastic digital architecture for recurrent (Hopfield like) neural networks is proposed. The main features and basic principles of stochastic processing are presented. The stochastic digital architecture is based on a chip with n fully interconnected neurons with pipeline, bit processing structure. For large applications a flexible way to interconnect many such chips is provided.

Introduction

The analog implementation of Hopfield neural network is of actual interest [5]. Due to the great complexity of the interconnections and to the presence of parasitic coupling path, analog recurrent networks are prone to follow incorrect trajectory or to oscillate. This reduces by an order of magnitude the number of neurons that can be built on a chip. In the same time, large applications require to interconnect many such chips. Due to the parasitic capacitance that distort the analog signals, this becoma another difficult task. A digital stochastic architecture avoids these problems. Here the signals are more easily passed between chips and are less modified by noise. By using a time-multiplexed structure, the connectivity is greatly reduced and so leads to flexible multi-chip systems. Recurrent networks operate by accumulating small changes into the neural state. This integrative process has a lowpass filtering effect, reducing also the inherent stochastic processing noise [2].

Following, an overview of the methods for information digital stochastic encoding, as well as some arithmetic computing elements, are presented. In the next sections our approach is detailed by providing: the algorithm, the block diagram of the proposed neural chip, and a detailed description of the synaptic and neural processors. Finally, the system expendability, reliability and reconfigurability are treated along with discussions on execution speed.

Digital stochastic encoding of information

A stochastic encoder is basically a tunable random pulse generator. The probability of occurrence for a pulse, i.e. the mean pulse rate, is controlled by the input to be encoded in such a way, that

$$p(x) = S_{in} / S_{max} \qquad (1)$$

In equation (1) $p(x)$ is the probability that the binary random pulse train assumes a value of 1 at a moment. S_{in} is the value to be encoded, and S_{max} represents the maximum possible value for the signal S. Thus the probability of a pulse in the pulse train is proportional to the normalized input signal. The basic circuit for encoding a digital signal (number) into a random pulse train with appropriate probability is shown in figure 1.

The number N is compared with a random number R, uniformly distributed over [Rmin,Rmax]. The output of the comparator will pulse if $R < N$, crating a stochastic firing signal X, whose mean is proportional to N provided N is in [Rmin,Rmax]. The stochastic encoding represents an analog signal mapping. By using non-weighted bits in a code of infinite word length, it is extremely noise-proof. In the same time it has an adaptive accuracy. As information is recovered through a pulse counting process, one can at any moment decide for a fast but imprecise or for a slow but accurate response. The computations are easily performed on such signals using space- and speed-efficient digital logic [4].

Arithmetic computing elements

The basic arithmetic computing elements used in this approach are: multiplication, counting/accumulation and linear/nonlinear transfer functions.

For example, if two statistically independent binary random pulses, x and y, with probabilities $p(x)$ and $p(y)$ are ANDed, the result has the probability:

$$p(r) = p(x) \text{ AND } p(y) = p(x)*p(y) \qquad (2)$$

That is, a multiplier in a stochastic architecture is a simple AND gate.

The easiest way to perform the accumulation function, which is equivalent with an integration operation, is to use a counter. For a neural processor, this will count the number of pulses which results from the multiplying operation between weights and neural outputs.

In simulating neural nets, the most time consuming operation is to apply the transfer function, which usually is a nonlinear one, to the neural state and obtain the neural output response. The use of stochastic encoding provide significant time gain in performing linear/nonlinear transfer functions. Returning to the digital-to-stochastic converter, one can see that the mean value of the output binary pulse train is:

$$\langle x \rangle = Pr\ \{R < N\} \qquad (3)$$

which is the cumulative distribution function (CDF) of R. If N is the value of the neural state $u(t)$, and x is the neural output pulse train, then the transfer function can be modified adjusting the probability distribution fr..ction (PDF) of the random number generator (see figure 1.a,b,c). While a uniform PDF gives a linear transfer function with hard limits, a PDF like in figure 1.c gives the more used sigmoidal transfer function.

The stochastic architecture

The equation governing the integration of charge in the Hopfield network is:

$$\frac{du_j}{dt} = \sum_{i=0}^{N-1} (T_{ij}V_i + I_j) \qquad (4)$$

If the time slice dt is much smaller than the main integration period, so that the capacitor voltages do not change too much, equation (4) can be approximated by:

$$u_j(ndt+dt) = u_j(ndt) + T_{ij}V_i(ndt)dt \qquad (5)$$

Therefore, in dt time, for each neuron j, the summation of only one product $T_{ij}V_i$, to the state u_j, is performed.

The architecture based on (5) has N neurons operating in parallel, at the clock frequency f_0, and an execution speed of N^*f_0 connections per second.

As long as the time-multiplexing implied by (5) has not prevented proper convergence or caused fault operation [1], equation (6) also holds true:

$$u_j(ndt+dt) = u_j(ndt) + \sum_{i=1}^{n}\sum_{r=1}^{b} (T_{r+(i-1)b,j}V_{r+(i-1)b}(ndt))\,dt \qquad (6)$$

where n*b < N. Thus, in dt time the summation of (n*b) products $(T_{ij}V_i)$, to the state u_j, is performed. In this way the state update speed can be enhanced n*b times.

The digital stochastic architecture proposed in this paper is based on equation 6. The basic building block is a chip with n fully interconnected neurons, operating in a pipe-line, bit serial manner on words of b bits length. The chip is depicted in figure 2. There are two types of processors: Synaptic Processors (SP) and Neural Processors (NP). Each synaptic processor, SP_{ij}, performs one product and two summations in parallel in a clock period. It contains a comparator Comp and a counter Count which are organized on bit serial, pipe-line structure (figure 3). The weight shift register (WSR) has a set of K registers on b bits word length. Skewing the outputs of the WSR and of the random number generator (RNG), every clock period T_0, a pulse with probability proportional to $|T_{ij}|$, results at the output of the comparator. The weight multiplication is performed by ANDing this signal with the neural output value received on V_i line. The result is added or substracted, to/from Count if T_{ij} is positive respective negative. This represents the summation over r index in equation 6. After b clock periods the content of Count is trasfered to the shift register (SR), and Count is reset. The value in SR is added serially with the value $u'_j(i-1)$, resulting $u'_j(i)$. The value $u'_j(i-1)$ is a partial sum of the neural state $u_j(t)$ computed in the previous cycle in the (i-1)th stage.

The neural processor NP_j has a bit serial adder, a digital-to-stochastic converter and a shift register which store the neural state value (figure 4). Every b clock periods, a new partial sum $u'_j(n)$ is added to the neural state $u_j(t)$. This process represents. the summation over h index in equation (6).

The lines V_i ,i=1,n, are internally connected to SP_{ij}, j=1.n, and are also used to interconnect chips forming large systems. External logic is needed for recovering the mean value of the signals which are the neural activity information.

Expendability, fault tolerance and speed

In large applications, which require a grate number of neurons, say N, K chips must be used (k = [N/n] + 1). By providing the SP with K_{max} weight registers, the neural network will be extensible to N_{max} neural processors. K_{max} is also the maximum number of chips that can be connected in a system. Such a system is organized around a n bits bus. In a period T_0, only one chip puts on the bus the output values of its neural processors. All chips will read these values, performing the state update function. This is done once, for each chip, after what the cycle is repeated.

This type of architecture, implying the connection to a unic bus, allows the number of neural processors to easily be changed by inserting or removing chips. In the same time, if a chip contains too many defective elements it may be bypassed by desselection. In this moment, an idle, unused, chip will replace the defective one.

The weight update speed can be evaluated taking into account that the total number of neurons in a system is $N = K*n$, and the number of weights is $W = (k*n)^2$. In each chip, for each neuron, (n*b) connections are computed in b clock periods (t_0), the speed being:

$$S = (K*n^2)/T_0 = n*N*f_0 \text{ connections/second} \qquad (7)$$

For example, a system with n = 100, K = 10 and f_0 = 100 MHz, has the execution speed $S = 10^{11}$ connections/second. This value is well above any reported implementations.

A time-multiplexed architecture performance parameter, is the neural state update speed (NUS). NUS is the number of products ($T_{ij}V_i$) added to the neural state in unit time. For the previous reported implementation [1], the NUS was equal to f_0 values per second. In our architecture NUS reach ($n*f_0$) values per second.

Conclusions

A stochastic digital architecture for networks of the recurrent type has been described in this paper. The low-pass filtering, integrative nature of these networks was well-suited for an implementation based on stochastic techniques built from entirely digital circuitry. The combination of all-digital signals and pipe-line, bit serial processors led to a system which could be spread across multiple chips. The reduced interconnectivity of the VLSI system made dynamic reconfigurability and fault tolerance easy to achieve.

References

1. D. VAN DEN BOUT, et.al.,"A Digital Architecture Employing Stochasticism for the Simulation of Hopfield Neural Nets' in IEEE Trans. on cir. and sys., vol.36, no. 5, pp 732 - 738.

2. D. VAN DEN BOUT, et.al.,"Scalable VLSI Implementation for NN" in Journal of VLSI Signal Processing,1,pp 367 - 385, 1990 Kluwer Academic Publishers.

3. J.J. HOPFIELD, "Neurons with graded response have collective computational properties like those of two-state neurons" in Proceedings of the National Academy of Sciences USA 81, May 1984, pp 3088-92.

4. ROBERT MASSEN,"Why do neural network researchers ignore stochastic computers?" in Parallel Processing in Neural Systems and Computers", R. Eckmiller, G. Hartman and G. Hanske, Eds., pp 23-26, Elsevier Science Publishers B.V.,1990.

5. STUART MACKIE,et.al.,"Implementations of neural network models in silicon" in NATO ASI Series, Vol.F41, "Neural Computers", R. Eckmiller and Ch.v.d. Malsburg, Eds., Springer-Verlag, 1988.

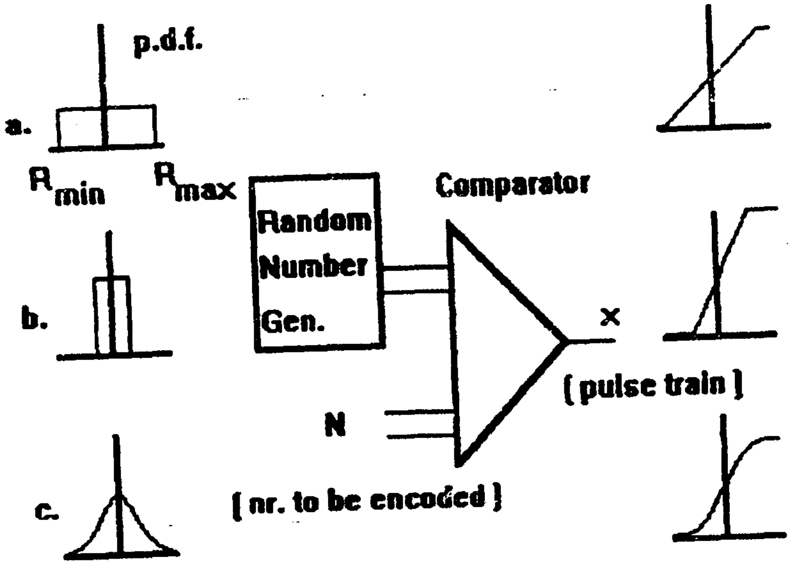


Figure 1 Digital to Stochastic Converter

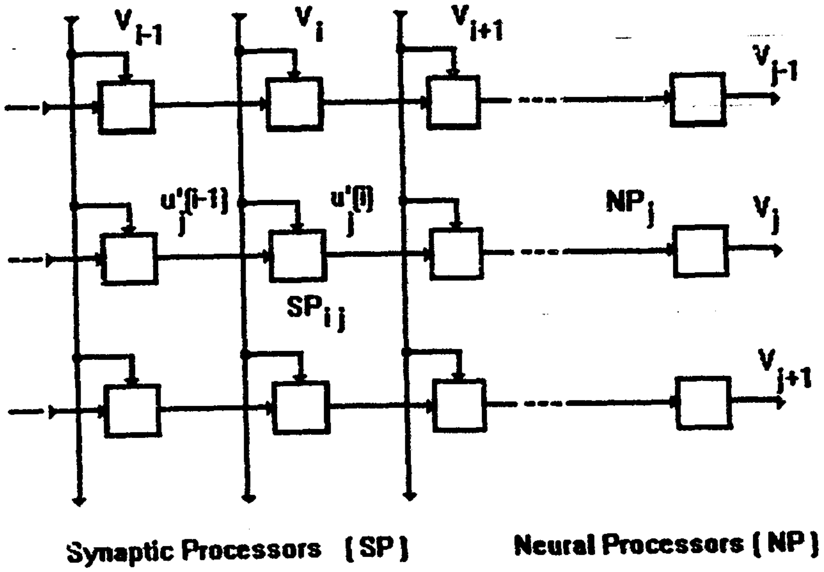
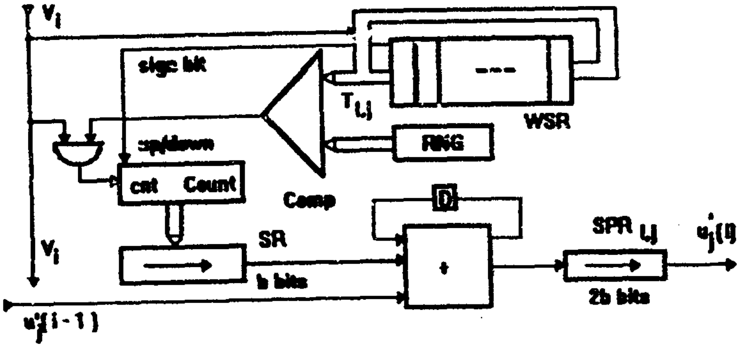


Figure 2 Neural Chip Architecture



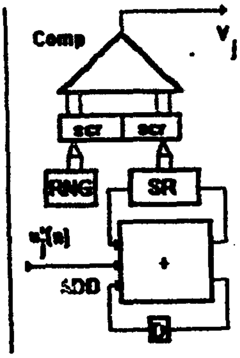


Figure 3 Synaptic Processor

Figure 4 Neural Processor

N93
29580
UNCLAS

524-67

161864

p-8

HIERARCHICAL MODEL OF MATCHING

Witold Pedrycz
Dept.of Electrical & Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
pedrycz@eeserv.ee.umanitoba.ca

Eugene Roventa
Dept.of Computer Science
Glendon College, York University
Toronto, Ontario, Canada M4N 3M6
Roventa@venus.Yorku.ca

Abstract The issue of matching two fuzzy sets becomes an essential design aspect of many algorithms including fuzzy controllers, pattern classifiers, knowledge-based systems, etc. This paper introduces a new model of matching. Its principal features involve: (i) matching carried out with respect to the grades of membership of fuzzy sets as well as some functionals defined on them (like energy, entropy, transom), (ii) concepts of hierarchies in the matching model leading to a straightforward distinction between "local" and "global" levels of matching; (iii) a distributed character of the model realized as a logic-based neural network.

Keywords matching, hierarchical model, local and global level of matching, logic-based neural networks

1.Introduction

Defining and handling problems of matching fuzzy sets (linguistic quantities) has become a domain of intensive research dating from the very emergence of fuzzy sets.The abundance of the matching methods available nowadays is evident. Different approaches stemming from measuring distances between membership functions, calculating possibility and necessitiy measures, using fuzzy measures and integrals, to name a few among them, give a good impression about their variety, cf. [4], [5], [6].

The proposed model embraces three new features being nonexistent or not fully addressed in the framework of the previous methodology. They are, however, important in dealing with fuzzy sets. One should stress that fuzzy sets form a collection of objects belonging to a given category to a certain degree. As such the grade of membership at $x_0 \in \mathbf{X}$ does not exist in isolation and is usually related (affected) by other membership values that the fuzzy set takes on in the neighbourhood of this point.This fact implies that this phenomenon should have a direct impact on the development of matching procedures.

Firstly, the two levels of hierarchy at which the matching process is carried out are distinguished: (i) a local level of matching dealing with the grades of membership of two fuzzy sets pertaining to the same element of the universe of discourse, and (ii) a global level of matching where all those "local" characteristics are summarized (aggregated).

Secondly, the local level of matching should also handle several criteria of matching not being exclusively restricted to the analysis of the grades of membership of the objects. Some other functionals defined over the membership values (like entropy,energy or transom) might be worth considering in this context.

The discussed model of matching is fully distributed and utilizes logic neurons [8], [9] to constructively accomplish matching at the indicated levels.

In the remainder of the paper we will consider fuzzy sets defined in a finite universe of discourse, say $\mathbf{X} = \{x_1, ..., x_n\}$. The discussion regarding the local level of matching will be covered in Section 2. In Section 3 we will proceed with the global level of matching showing how different elements of X interact within the process of matching. The learning algorithm leading to parametric adjustments of the connections in the model will be studied in Section 4.

2. The pointwise level of matching of fuzzy sets

Let us discuss two grades of membership at a certain element of **X**, say $a = A(x_i)$, $b = B(x_i)$ where A and B are fuzzy sets. The general questionarises: why are these fuzzy sets similar or what makes them different? First of all it is likely that a very preliminary answer to this problem can be formulated by studying the corresponding values of the membership functions of A and B.They are usually deemed essential in expressing similarity between fuzzy sets.
One among existing alternatives useful in describing similarity of fuzzy sets could be the use of the following equality index cf. [7]:

$$a \equiv b \leftrightarrow a = b = \frac{1}{2}[(a \varphi b) \wedge (b \varphi a) + (\overline{a} \varphi \overline{b}) \wedge (\overline{b} \varphi \overline{a})] \qquad (1)$$

where the φ-operation (pseudocomplement) is defined as follows:

$$a \varphi b = \sup\{c \in [0,1] \mid atc \le b\}$$

and "t" denotes a triangular norm while "$\wedge$"stands for minimum. The equality index attains 1 if and only if the arguments are equal, $a = b$. It should be stressed that the values produced by the equality index are not context sensitive,viz.this index produces the same result once a mutual position of the arguments is the same.This means, for instance, that $0.1 \equiv 0.1 = 1$ and $0.9 \equiv 0.9 = 1$. This could cause undesired lack of discriminatory properties of this definition. On the contrary, it could be propitious to discriminate between situations where matching achieved for the higher membership values such as 0.9 and 0.8 is more significant than the one reported for the lower values ,say 0.05 and 0.2. One of feasible solutions to this deficiency would be to perform (1) not only on the membership grades but also on their functionals. We will study three well-known families of these membership functionals:

- energy-type functionals [1], [2] compute values of a certain monotonically nondecreasing function defined over the original membership values,say ψ_1: $[0,1] \to [0,1]$ where ψ_1 is a monotonically nondecreasing function. For instance, one can refer to polynomial-type energy functionals of the form $\psi_1(u) = u^p$, $p > 1$;
- entropy-type functionals, cf. [1] [2],are defined as mappings ψ_2: $[0,1] \to [0,1]$ such that

  (i) ψ_2 is monotonically increasing over [0, 1/2] and

  (ii) ψ_2 is monotonically decreasing over [1/2, 1].

  Moreover one assumes that the functionals attain maximum at 1/2, $\psi_2(1/2)=1$.
- transom-like functionals [10], [11] remind the functionals of the first class .The modification is such that low and high grades of membership (i.e., the values lying around 0 and 1) are ignored. One can characterize these functionals as ψ_3: $[0,1] \to [0,1]$, such that

  $\psi_3(u) = 0$, if $u \le \alpha$,

  $\psi_3(u) = \psi_1(u)$, if $u \in (\alpha, \beta)$, and

  $\psi_3(u) = 0$, if $u \ge \beta$

  where α and β are threshold levels.

The examples of these functionals expressed with the aid of linear or piecewise linear relationships are included in Fig. 1.

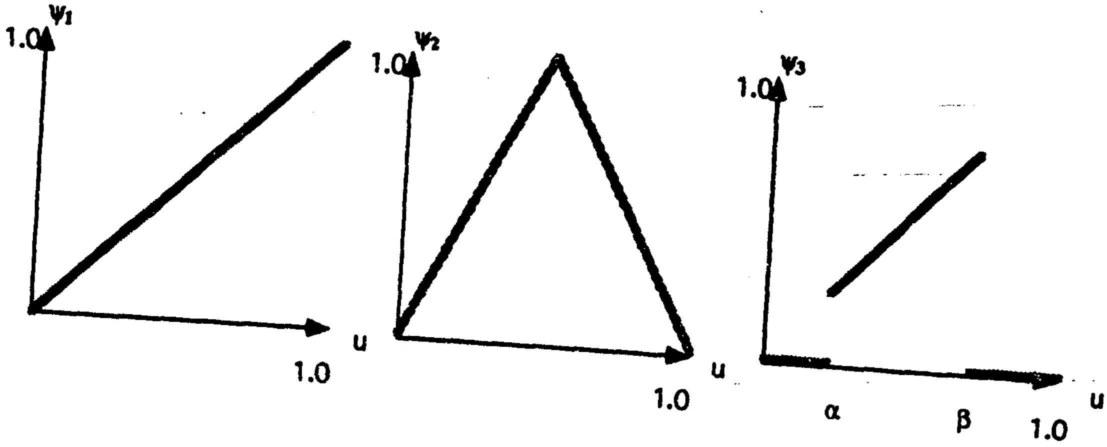


Fig.1.Examples of functionals ψ_1,ψ_2,ψ_3

Here we have:

$\psi_1(u) = u,\ u \in [0, 1]$

and $\psi_2(u) = 2u,\ u \in [0, 1/2]$ and $\psi_2(u) = 2(1 - u),\ u \in [1/2, 1]$

$\psi_3(u) = \psi_1(u)[\mathbf{1}(u - \alpha) - \mathbf{1}(u - \beta)]$

where **1** denotes the unit step function i.e. $\mathbf{1}(u - \zeta) = 0,\ u \le \zeta$ and $\mathbf{1}(u - \zeta) = 1,\ u > \zeta$.

Generally speaking, the result of this "local" (pointwise) matching can be summarized (aggregated) in a logical way as:

$$z = \left[\left(\psi_1\left((A(x_i)) \equiv \psi_1(B(x_i))\right)\right) \text{OR } w_1\right] \text{AND} \left[\left(\psi_2\left((A(x_i)) \equiv \psi_2(B(x_i))\right)\right) \text{OR } w_2\right] \text{AND} \ldots \text{AND} \left[\left(\psi_p(A(x_i)) \equiv \psi_p(B(x_i))\right) \text{OR } w_p\right] \tag{2}$$

where w_i, i = 1,2,...,p are weights (connections associated with the consecutive degrees of equality, while "z" denotes an overall level of matching obtained at x_i. The weights modulate influence of the individual components on the local result of matching.

From a structural point of view (2) can be treated as an AND logical neuron, cf. [9], see also Fig. 2.

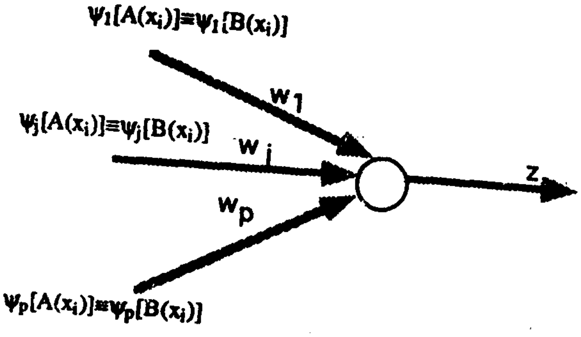


Fig.2.AND logic-based neuron

The appropriate values of the connections can be derived through supervised learning. We will discuss this issue in great detail in Section 4.

In order to emphasize the local character of matching and explicitly indicate the arguments standing there (elements of **X**) we will introduce a two-variable predicate MATCH_LOCAL(x_i, x_j) which is defined as follows,

$$\text{MATCH\_LOCAL}(x_i, x_j) = \Big[\big(\psi_1(A(x_i)) \equiv \psi_1(B(x_j))\big) \text{ OR } w_1\Big] \text{ AND } \ldots \text{ AND } \Big[\big(\psi_p(A(x_i)) \equiv \psi_p(B(x_j))\big) \text{ OR } w_p\Big]$$

i,j = 1,2,...,n

3.Global level of matching

When it comes to the global level of matching involving all the elements (pairs) for which the local matching operation has been accomplished we can think of the following model,

$$y = \text{OR } [\text{MATCH\_LOCAL } (x_i, x_j) \text{ AND } v_{ij}] \qquad (3)$$

i,j = 1,2,...,n

where v_{ij}, i,j ∈ [0,1] are connections modelling the influence the results of the local matching have on the global level.

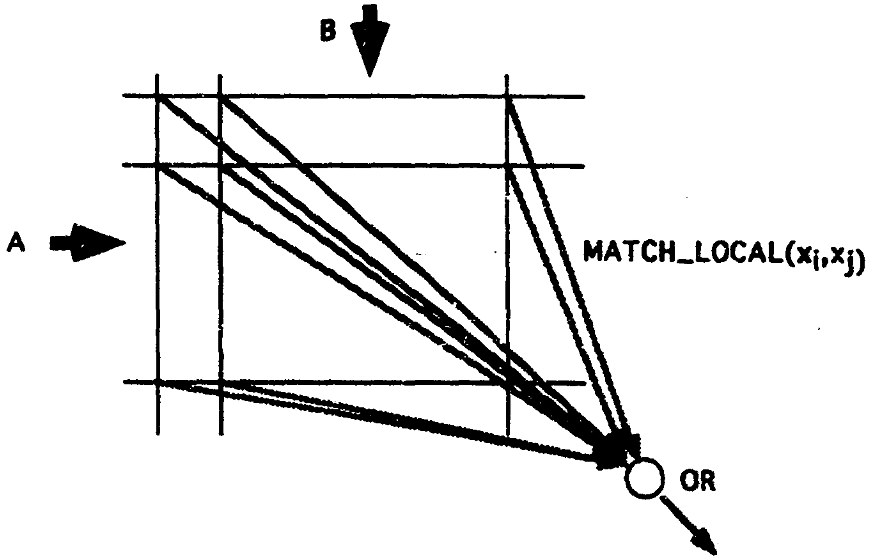


Fig.3.Overall matching model

The complete structure composed of the processing units described by (3) is given in Figure 3. The grid of points shown there is formed by considering a Cartesian product of **X**'s.

The entire model can be viewed as a heterogeneous OR-AND logic neural network, see [9]. The compact notation applied to it will then look like this,

$$y = \text{MATCH}(A, B) = \underset{(x_i,\, x_j) \in \mathbf{X}x\mathbf{X}}{\text{OR}} [\text{MATCH\_LOCAL}(x_i, x_j) \text{ AND } v_{ij}] \qquad (4)$$

where the OR operation pertains to the arguments of the Cartesian product **X**x**X**. The AND and

OR operations are modelled by triangular norms (t- and s-norms, respectively). This implies the following system of relationships,

$$\text{MATCH(A,B)} = \underset{i,j\ =\ 1,2,\ldots,p}{S} [\text{MATCH\_LOCAL}(x_i, x_j)\ t\ v_{ij}] \qquad (5)$$

$$\text{MATCH\_LOCAL}(x_i, x_j)) = \underset{l\ =\ 1,2,\ldots,p}{T} [\psi_l(A(x_i)) \equiv \psi_l(B(x_j)))\ s\ w_l] \qquad (6)$$

Expressions (5) - (6) form a basic distributed and hierarchical model of matching.

In the foregoing section we will investigate a problem of parametric learning in (5) - (6). This will include the connections v_{ij} and w_l.

4. Specializing the model of matching- a parametric learning in the network

Since the structure of the model has been already developed ,now one has to determine its connections $w = [w_l]$, $l = 1,2,\ldots p$ and $v = [v_{ij}]$ $i,j = 1,2,\ldots,n$ .This is carried out on the basis of a training set of data. It consists of pairs of fuzzy sets A_k, B_k and associated results of matching reported there. Denote them by t_k, $k = 1,2,\ldots,N$. Usually we can concentrate on a simple scenario in which $t_k \in \{0,1\}$. If $t_k = 0$ the corresponding pair (A_k, B_k) delineates two fuzzy sets which are different. On the other hand, if $t_k = 1$, A_k and B_k are viewed as being similar.

The learning (adjustments) of the connections is completed in the supervised mode. One presents A_k and B_k to the model and compares the obtained result MATCH(A_k, B_k) with t_k: if these are different then w and v have to be modified to reduce this difference. A convenient performance measure guiding the adjustments of the connections (parametric learning) is a sum of squared errors.

$$Q = \sum_{k=1}^{N} [t_k - \text{MATCH}(A_k, B_k)]^2 \qquad (7)$$

Then a standard Newton-like method is exploited to produce the required modifications of w and v,namely,

$$w = w - \eta \partial Q/\partial w$$

$$v = v - h \partial Q/\partial v$$

where η is a learning rate controlling a speed of changes of w and v.

The derivatives are computed in a standard way,

$$\frac{\partial Q}{\partial v_{ij}} = -2\sum_{k=1}^{N} [t_k - \text{MATCH}(A_k, B_k)]\, \frac{\partial \text{MATCH}(A_k, B_k)}{\partial v_{ij}}$$

$$\frac{\partial \text{MATCH}(A_k, B_k)}{\partial v_{ij}} = \frac{\partial}{\partial v_{ij}}\Big[\underset{i1,\ j1\ =\ 1,2,\ldots,n}{S} [\text{MATCH\_LOCAL}(x_{i1}, x_{j1})\ t v_{i1,j1}]\Big] =$$

$$= \frac{\partial}{\partial v_{ij}} [[\text{MATCH\_LOCAL}\ (x_i, x_j)\ t v_{ij}] s S_1]$$

where S_1 abbreviates the result obtained for the computations carried over the elements of XxX different from x_i and x_j. The detailedcalculations can be pursued further upon specification of the triangular norms. For the connections w we derive similarly,

$$\frac{\partial Q}{\partial w_1} = -2\sum_{k=1}^{N} [t_k - \text{MATCH}(A_k, B_k) \frac{\partial \text{MATCH}(A_k, B_k)}{\partial w_1}$$

and

$$\frac{\partial \text{MATCH}(A_k, B_k)}{\partial w_1} = \sum_{i,j=1,2,\ldots,n} \frac{\partial \text{MATCH}(A_k, B_k)}{\partial \text{MATCH\_LOCAL}\,(x_i, x_j)} \times \frac{\partial \text{MATCH\_LOCAL}(x_i, x_j)}{\partial w_1}$$

Subsequently we conclude

$$\frac{\partial \text{MATCH}(A_k, B_k)}{\partial \text{MATCH\_LOCAL}\,(x_i, x_j)} = \frac{\partial [S_2\, t\text{MATCH\_LOCAL}(x_i, x_j)\, tv_{ij}]}{\partial \text{MATCH\_LOCAL}\,(x_i, x_j\,)}$$

with

$$S_2 = \underset{i1,\ j1,\ i1 \neq i,\ j1 \neq j}{S} [\text{MATCH\_LOCAL}\,(x_{i1}, x_{j1})\ tv_{i1,j1}]$$

and finally

$$\frac{\partial \text{MATCH\_LOCAL}(x_i, x_j)}{\partial w_1} = [T_1 t(\psi_1(A(x_i)) \equiv \psi_1(B(x_j)))\ sw_1]$$

$$T_1 = \underset{l_l \neq 1}{T} [\psi_{11}(A(x_{l_l})) \equiv \psi_{12}(B(z_{l_l})))\ sw_{l_l 1}]$$

The following example serves as an illustrative material showing how the hierarchical model of matching can be developed

Example The fuzzy sets used in this simulation experiment are given below:

	x_1	x_2	x_3		x_1	x_2	x_3	Matching t_k
A_1 =	[0.95	0.71	0.2]	B_1 =	[0.4	0.9	0.05]	1.0
A_2 =	[0.1	0.3	0.58]	B_2 =	[0.15	1.0	0.01]	0.0
A_3 =	[0.25	0.4	1.0]	B_3 =	[0.5	1.0	0.7]	1.0
A_4 =	[0.45	1.0	0.75]	B_4 =	[1.0	0.2	0.45]	0.0

This synthetic data set includes some pairs of fuzzy sets exhibiting equality (t_k=1) and difference between A_k and B_k's (t_k=0). We will consider the functionals ψ_1, ψ_2 and ψ_3 shown in Fig.1(for the transom functional we set $\alpha = \beta = 0.1$) .

The network is described with the use of (5) - (6), the performance index is given by (7) and $\eta = 0.2$.THe process of learning is visualized in Fig.4.

The results generated by the network are given below

k	Matching (t_k)	MATCH (A_k, B_k)
1	1.0	0.71

2	0.0	0.33
3	1.0	0.64
4	0.0	0.43

Even though theoutcomes in these two columns are numerically different,they become qualitatively equivalent after thresholding applied to the results produced by the matching model.Let us define a threshold operation

$$\tau(a, \lambda) = 1 \text{ if } a \geq \lambda \text{ and } \quad \tau(a, \lambda) = 0 \text{ if } a < \lambda,$$

$a, \lambda \in [0,1]$.

where λ stands for the threshold value.One can easily verify that for all λ from (0.43,0.64) the results produced by the model (after thresholding) are equivalent to the these included in the training set.

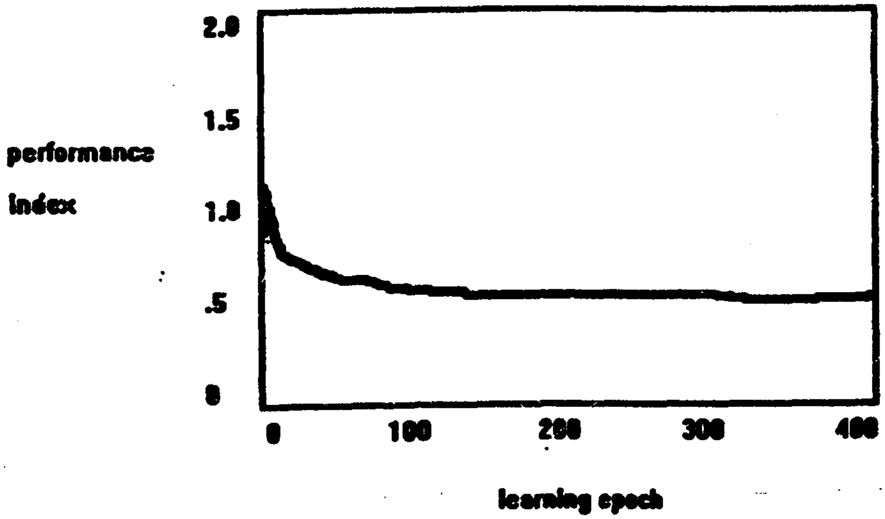


Fig.4.Results of learning in the network

5. Conclusions

We have developed a distributed model of matching of fuzzy sets utilizing AND and OR basic computing elements. It has been shown that they carry out "local" matching which is realized at the level of each element of the universe of discourse and includes both grades of membership as well as some of their functionals. The global level aggregates them in a disjunctive form.

The idea of the distributed logical processing can be also found useful in developing models of fuzzy set connectives or designing non-pointwise decision-making procedures.

Acknowledgement

Support from the Natural Sciences and Engineering Research Council of Canada for both authors and MICRONET for the first author is gratefully acknowledged.

6.References

1. A. De Luca, S. Termini, "A definition of nonprobabilistic entropy in the setting of fuzzy sets", Information and Control, 20, 1972, 301-312.
2. A. De Luca, S. Termini, "Entropy and energy measures of a fuzzy set". In: Advances in Fuzzy

Set Theory and Applications, M.M. Gupta, R.K. Ragade, R.R. Yager eds., North Holland, Amsterdam, 1979.

3. D. Dubois, H. Prade, "Fuzzy sets and statistical data", European J. of Operational Research, 25, 1986, 345-356.
4. D. Dubois, H. Prade, C. Testemale, "Weighted fuzzy pattern matching", Fuzzy Sets and Systems, 28, 1986, 313-352.
5. D. Dubois, H. Prade, 'Théorie des Possibilités, Masson, Paris, 1985.
6. K. Hirota, W. Pedrycz, Handling fuzziness and randomness in processes of matching of fuzzy data", Proc. 3rd IFSA Congress, Seattle, August 1986.
7. W. Pedrycz, "Direct and inverse problems in comparison of fuzzy data", Fuzzy Sets and Systems, vol. 35, 2, 1990, 223-2367.
8. W. Pedrycz, "Neurocomputations in relational systems", IEEE Trans. on Pattern Analysis and Machine Intelligence, 13, 1991, 289-296.
9. W. Pedrycz, "Fuzzy neural network with reference neurons as pattern classifier", IEEE Trans. on Neural Networks,1992 (in press).
10. E. Roventa, "On the degree of fuzziness of a fuzzy set", Fuzzy Sets and Systems, vol. 36, 2, 1990, 259-264.
11. E. Roventa, "On a class of measures of fuzziness", J.Math.Analysis and Appl. (in press).
12. B. Schweizer, A. Sklar, Probabilistic Metric Spaces, North Holland, Amsterdam, 1983.

A Conjugate Gradients/Trust Regions Algorithm for Training Multilayer Perceptrons for Nonlinear Mapping

Raghavendra K. Madyastha   Behnaam Aazhang
Department of Electrical and Computer Engineering, Rice University, Houston, TX
Troy F. Henson   Wendy L. Huxhold
IBM Corporation, Houston, TX

October 13, 1992

Abstract

This paper addresses the issue of applying a globally convergent optimization algorithm to the training of multilayer perceptrons, a class of Artificial Neural Networks. The multilayer perceptrons are trained towards the solution of two highly nonlinear problems: i) Signal detection in a multi-user communication network and ii) Solving the inverse kinematics for a robotic manipulator. The research is motivated by the fact that a multilayer perceptron is theoretically capable of approximating any nonlinear function to within a specified accuracy. The algorithm that has been employed in this study combines the merits of two well known optimization algorithms, the Conjugate Cradients and the Trust Regions Algorithms. The performance is compared to a widely used algorithm, the Backpropagation Algorithm, that is basically a gradient-based algorithm, and hence, slow in converging. The performances of the two algorithms are compared in terms of the convergence rate. Furthermore, in the case of the signal detection problem, performances are also benchmarked by the decision boundaries drawn as well as the probability of error obtained in either case.

I Introduction

Artificial Neural Networks (Neural Nets for short) are densely interconnected layers of relatively simple processing units called *nodes*, that are interconnected through links called *weights*, ($\underline{\omega}$ represents the *weight vector*). The output of any node in a layer is a nonlinear function of a weighted sum of inputs from nodes in the previous layer. Due to the nonlinear characteristics of these networks, they are used for a wide variety of nonlinear mapping problems. This paper deals with two specific applications : detection of a single user's signal in a multi-user communication channel and solution of the inverse kinematics for a robotic manipulator. The neural net used in these problems is the multilayer perceptron.

Multilayer perceptrons are a class of feed-forward artificial neural networks [1, 2, 3] with one or more layers (termed *hidden layers*) between the inputs and outputs. Their use in this context is based on the fact that multilayer perceptrons with a single hidden layer are theoretically capable of approximating any nonlinear function to a desired accuracy [4]. The general classification/mapping problem can be reduced to solving an optimization problem as shown below

$$\underline{\omega}_* = \arg \min_{\underline{\omega} \in \Re^n} \epsilon(\underline{\omega}) : \ \Re^n \longrightarrow \Re . \tag{1}$$

The optimization algorithm used to calculate $\underline{\omega}_*$ that solves (1) is termed the *training algorithm* of the multilayer perceptron. The *error function* ϵ is typically taken to be an average of the sum of the squares of the differences between the desired and actual (produced by the neural net) outputs to given inputs

$$\epsilon(\underline{\omega}) = \frac{1}{P} \sum_{p=1}^{P} \sum_{i=1}^{M_L} (\hat{d}_i(p ; \underline{\omega}) - d_i(p))^2 , \tag{2}$$

where $d_i(p)$ is the i^{th} component of the the desired output vector, i.e., the desired output at the i^{th} node (of M_L output nodes) corresponding to the p^{th} input pattern, $\hat{d}_i(p; \underline{\omega})$ represents the actual output and $p = 1, \ldots, P$ represent the total number of training patterns presented to train the neural network.

General optimization problems as in (1) have no analytic solutions and hence, iterative optimization schemes that yield a series of converging approximations to $\underline{\omega}_*$ are employed to solve (1). The focus of this research is the development of an efficient training algorithm that performs "significantly better" than the widely employed *backpropagation* algorithm [1, 3, 5, 6]. Since the backpropagation algorithm is a gradient-based algorithm (it is based on the steepest descent algorithm) it exhibits very poor convergence properties[1], being at best linearly convergent [7, 8, 9]. We investigate an optimization algorithm that combines the merits of two well known optimization algorithms : the ***trust region algorithms*** and the *conjugate gradient algorithm*. The resulting algorithm, termed the Conjugate gradient-Trust region (CGTR) algorithm has been shown to exhibit superlinear convergence properties [10, 11]. The CGTR algorithm significantly outperforms the backpropagation algorithm in the applications considered.

II Multilayer Perceptrons

Multilayer perceptrons form a particular class of neural networks and are capable of approximating any nonlinear measurable functions. Specifically, it has been shown by Hornik, Stinchcombe and White [4] that a two-layer perceptron, i.e., a perceptron with an input layer, one hidden layer and an output layer of nodes, is sufficient to achieve this approximation. This capability has been exploited in various fields including speech recognition, signal and pattern classification and universal approximation (see references in [1]). With reference to a single-user detection problem in a multi-user communication channel, wherein the optimal decision boundary has been shown to be a highly nonlinear curve in the signal space [12, 13], multilayer perceptron receivers have been observed to perform better than conventional techniques [14]. In this study we apply the multilayer perceptron, which is trained using the CGTR algorithm as opposed to the conventional backpropagation, to the two specific problems at hand, signal detection and nonlinear mapping.

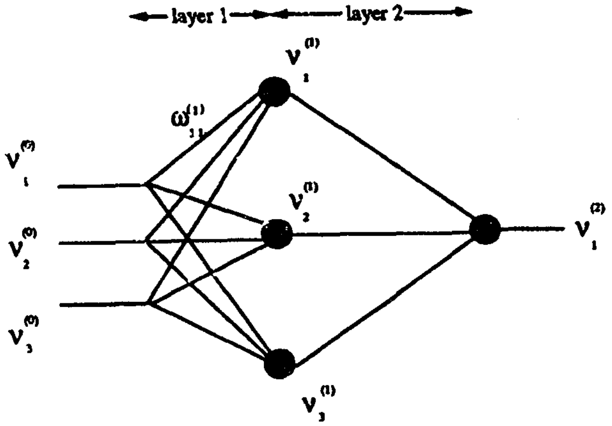


Figure 1: Typical Structure of a 2-Layer Neural Perceptron

These networks consist of an input layer of nodes, one or more layers of hidden (i.e., intermediate) nodes and a layer of output nodes (Figure(1)). The nodes in a given layer are connected to all the nodes of the next (upper) layer. Therefore in an L-layer perceptron, the output of the i^{th} node of the ℓ^{th} layer takes the value

$$\nu_i^{(\ell)} = g(\sum_{j=1}^{M_{\ell-1}} \omega_{ji}^{(\ell)} \nu_j^{(\ell-1)} - \omega_{0i}^{(\ell)}), \quad i = 1, 2, \cdots, M_\ell, \quad \ell = 1, 2, \cdots, L, \tag{3}$$

where M_ℓ denotes the number of nodes in the ℓ^{th} layer, $\omega_{ji}^{(\ell)}$ denotes the weight associated with the connection between the j^{th} node of the $(\ell-1)^{st}$ layer to the i^{th} node of the ℓ^{th} layer and $\omega_{0i}^{(\ell)}$ is the corresponding threshold. The function $g(\cdot)$ is the nonlinear transformation at the output of the i^{th} node of the ℓ^{th} layer, called the *activation* or *squashing* function. In this model, $\nu_j^{(0)}$ represents the j^{th} input to the network and M_0 is the total number of inputs. The measure of the error $\epsilon(\cdot)$ that arises naturally from the network configuration is the average sum of squared errors which is calculated as in (2) with $\hat{d}_i(p; \underline{\omega}) = \nu_i^{(L)}(\underline{\omega})$ representing the actual perceptron output.

[1] For details regarding the different types of convergence see [7, 8].

III Conjugate Gradients/Trust Regions Training Algorithm

The CGTR algorithm is a nested combination of the trust regions algorithm and the conjugate gradients algorithm that attempts to circumvent the limitations of both schemes. In order to illustrate the properties of the proposed training algorithm, a brief overview of iterative optimization algorithms is presented below, with emphasis on the trust regions model and the conjugate gradient algorithm.

The basic optimization problem can be formulated as in (1), where $\epsilon(\underline{\omega})$ is a twice continuously differentiable function of $\underline{\omega}$. The main strategy in most optimization algorithms is to approximate the nonlinear function $\epsilon(\underline{\omega})$ about the point $\underline{\omega}_k$ by a second order Taylor series expansion, called the *quadratic model* of ϵ at $\underline{\omega}_k$

$$m_k(\underline{\omega}_k + \underline{s}) = \epsilon(\underline{\omega}_k) + \nabla\epsilon(\underline{\omega}_k)^T \underline{s} + \frac{1}{2}\underline{s}^T H \underline{s}\,, \tag{4}$$

where H represents the Hessian matrix of ϵ i.e., $[H]_{i,j} = \partial^2\epsilon/\partial\omega_i\partial\omega_j$. One optimization scheme is to successively minimize the function along the *steepest descent* direction, i.e., the negative gradient at each point. This algorithm has been found to possess extremely slow (linear) convergence properties. The prevalent *Backpropagation* algorithm is based on the steepest descent algorithm.

An alternative class of algorithms that are extremely robust with respect to the wide variety of functions to which they are applicable are the trust region algorithms. The main idea behind these methods is that the given nonlinear function is approximated fairly accurately by a Taylor's series quadratic model only in some *region* around the current point. This leads to the following formulation of the optimization problem

$$\min_{\underline{s} \in \mathbf{R}^n} m_k(\underline{\omega}_k + \underline{s}), \quad \text{subject to } \|\underline{s}\| \leq \delta_k\,, \tag{5}$$

where $m_k(\underline{\omega}_k + \underline{s})$ is as defined in (4), $\underline{s}$ is the step taken and δ_k is a parameter that can be interpreted as an estimate of how far we "trust" $m_k(\underline{\omega}_k + \underline{s})$ to accurately model the actual function $\epsilon(\underline{\omega})$ in a neighborhood about $\underline{\omega}_k$. The parameter δ_k is accordingly called the *trust radius*. The trust region algorithm can then be succinctly stated as minimizing $m(\underline{\omega} + \underline{s})$ over a compact domain in $\underline{s}$ at each iteration. This problem has been shown to have a unique solution [7, 15, 16] for arbitrary H, but is numerically intractable.

The conjugate gradients algorithm is an algorithm that arrives at the minimizer of a positive definite n-dimensional quadratic function (i.e., the Hessian H is positive definite) in at most n steps [9, 17, 18, 19], taken along mutually *H-conjugate*[2] directions. It is a computationally simple and elegant algorithm that has minimal storage requirements. The rate of convergence of the conjugate gradients algorithm is found to depend on the condition number κ of H [9, 17, 20, 21]. Therefore, the convergence rate could be enhanced by suitably modifying the Hessian, clustering its eigenvalues and thereby decreasing κ. This technique, known as *preconditioning*, is achieved by affecting a linear transformation of the variable space. Notwithstanding the attractive features of the conjugate gradients algorithm, it is shown to be numerically unstable when applied to non-positive definite functions [20, 17, 18, 22]. Therefore it has to be used in conjunction with a method that allows for indefinite Hessians.

The proposed CGTR algorithm effectively combines the merits of both the above stated algorithms [10, 11]. The problem addressed is as posed by (5), with the imposed constraint being $\|\underline{s}\|_C \leq \delta_k$, where $\|\underline{s}\|_C^2 = \langle \underline{s}\,, C\underline{s} \rangle$, C being the preconditioning matrix. The conjugate gradients algorithm is *embedded* in the trust region model and serves to arrive at the minimizer of $m_k(\underline{\omega}_k + \underline{s})$ at each iterate $\underline{\omega}_k$, while the trust region part decides whether a particular $\underline{\omega}_k$ reached is a "suitable" point or not. The trust region algorithm governs the global convergence and thus enables the CGTR algorithm to effectively deal with non-positive definite Hessians. This algorithm has been theoretically shown to be superlinearly convergent to a minimizer of the nonlinear function ϵ [10, 11].

IV Applications

In view of the deficiencies in the existing traaining algorithms, there is a need for the development of faster and more robust algorithms. This paper discusses the application of the CGTR algorithm to the training of multilayer perceptrons for signal classification and nonlinear mapping (inverse kinematics) problems based on their universal approximation capabilities [4]. The multilayer perceptron is trained in both cases by presenting a set of input data and minimizing the resultant error function between the desired and actual outputs of the network to arrive at an optimal weight configuration $\underline{\omega}_*$. The input-output set in the signal detection scenario consists of P pairs of the sampled received signal vector and the corresponding classification, while in the solution of the inverse kinematics of a robotic manipulator, the input-output set consists of P pairs of end effector (see Section IV.2) coordinates and the corresponding joint angles. Figure 2

[2] Two directions p_i and p_j are said to be H-conjugate directions if $p_i^T H p_j = 0 \forall i \neq j$.

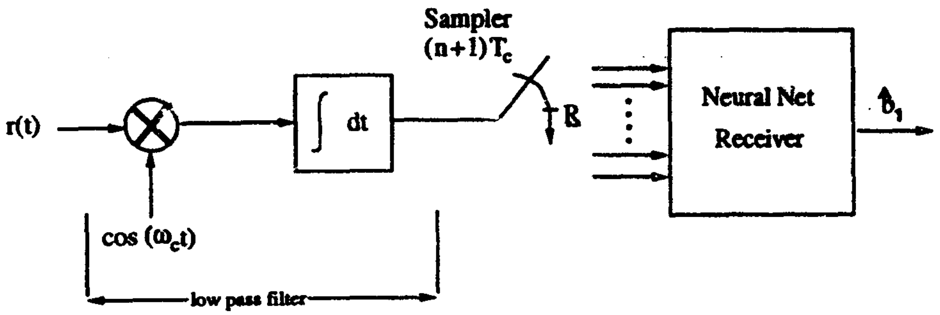


Figure 2: Structure of a Neural Net Receiver for Single User Detection in Multiuser Channels.

IV.1 Single User Detection in a Multi-user Communication Channel

We investigate the feasibility of using multilayer perceptrons with the proposed training algorithm for the detection of signals transmitted by a single user in a multi-user channel with additive Gaussian noise. The neural net receiver is configured to demodulate the particular user's signal in the presence of other interfering signals. This is shown to be equivalent to approximating a nonlinear function, the optimum decision boundary [12, 13]. Figure 3

In the general multiple-access communication network [14], K transmitters are assumed to share a radio band in time and code domains. A particular user's transmitted signal is a binary signal set derived from the set of coded waveforms assigned to that user. We assume that we are interested in the demodulation of the first user's information packet. The signal at the receiver is the sum of the K transmitted signals in additive channel noise (which is assumed to be Gaussian here).

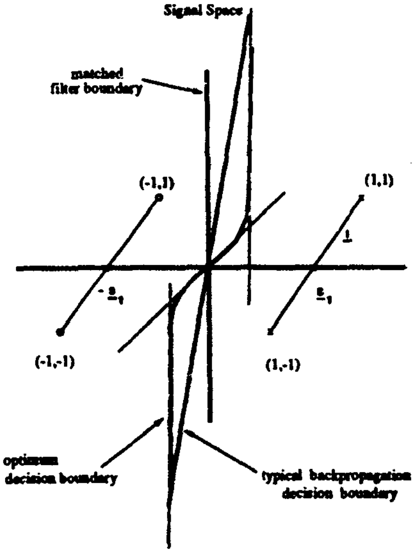


Figure 3: Optimum Decision Boundary for the Detection of a Single User in a 2-user Channel

The sampled input vector $\underline{R}$, to the neural net receiver (see Figure 2) can be written so that the demodulation of the first signal is viewed as the following classification problem:

$$\begin{aligned} H_0 &: \underline{R} = +A_1'\underline{a}^{(1)} + \underline{\eta} + \underline{I} \\ H_1 &: \underline{R} = -A_1'\underline{a}^{(1)} + \underline{\eta} + \underline{I}, \end{aligned} \tag{6}$$

where $\underline{a}^{(1)}$ is the spreading code vector of the first user and $\underline{\eta}$is a length-N vector of filtered Gaussian noise samples. In this setting, $\underline{I}$ represents the multiple-access interference vector, i.e., the interference due to the presence of the other transmitted signals. The optimum decision boundary for the general single-user detection problem in the presence of interfering users has been found to be a highly nonlinear surface in the signal space[13]. Therefore conventional matched filter techniques, which generate only linear detection boundaries (see Figure 3) fail to accurately demodulate the desired user's signal.

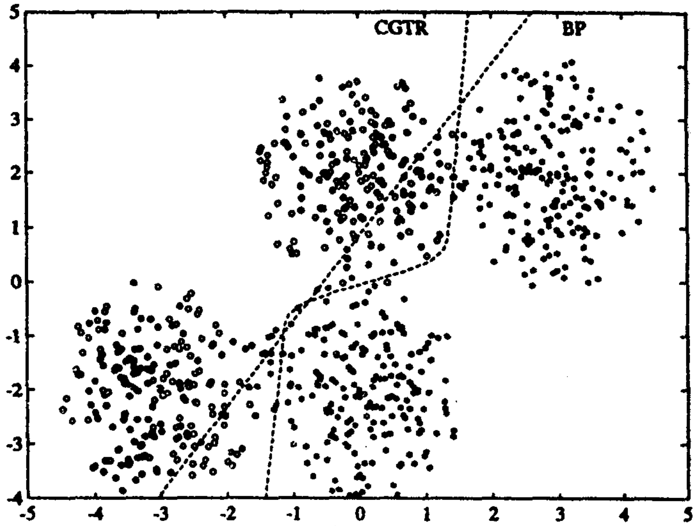


Figure 4: Decision boundaries drawn after training with the CGTR and BP algorithms.

Performance Analysis

To illustrate the potential of the multi-layer perceptron for signal detection, a relatively simple example involving the detection of a single user's signal in presence of only one other interfering user is considered. The network used for this problem is a two-layer perceptron with three nodes in the middle layer. This is based on work done by Aazhang, Orsak and Paris [14] who have conjectured that, since the optimum decision boundary can be approximated by three straight lines, three nodes in the middle layer are sufficient for near-optimum demodulation.

Training of the multi-layer perceptron is performed by presenting a fixed number of input vectors to the network and specifying the corresponding desired outputs. The error function obtained is then minimized with respect to the network weights using the CGTR method. In the case of the signal classification problem, the P input data represent observations of $\underline{R}$, i.e., actual signal locations with additive noise. The relevance of using signal with noise as training data lies in the fact that in a practical application the neural net receiver would be receiving noisy data and would have to detect the presence of a particular user in the presence of additive noise as well as interfering signals. Therefore, training the multilayer perceptron with noisy data makes the detector insensitive to perturbations in the incoming signals.

The performance of the multilayer perceptron trained by a particular algorithm in the case of the detection of a single user in the presence of interfering users, is assessed by the proximity of the decision boundary drawn by the multilayer perceptron to the optimum and the average probability of mis-classification. Figure 4 shows the decision boundaries drawn by the neural net receiver trained with the CGTR and backpropagation algorithms. As can be seen, the decision boundary drawn after training with the CGTR algorithm closely approximates the optimum decision boundary, while the neural net trained with the backpropagation algorithm is only able to linearly approximate this nonlinear function. Further comparisons of performance can be made by observing the plots of the probability of error for the receiver versus the ratio of the signal-noise-ratios of the two signals after training with both the algorithms, as seen in Fig 5. The first plot in Fig 5 depicts the probability of error (Pe) graphed against the ratio of SNR1 (signal-noise-ratio of user 1) to SNR2 (signal-noise-ratio of user 2), with SNR2 fixed at 6dB. The

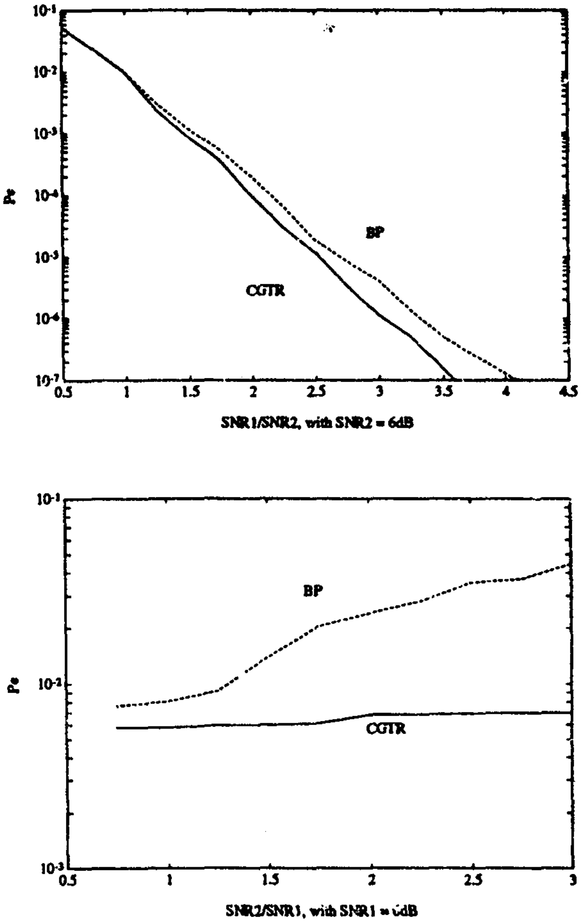


Figure 5: Probability of error vs ratio of the SNR's of the two signals.

second piot depicts Pe graphed against the ratio of SNR2 to SNR1, with SNR1 fixed at 6dB. As can be seen in both plots, the receiver trained with the CGTR algorithm yields a lower probability of error compared to the receiver trained with backpropagation.

IV.2 Inverse Kinematics for a Robotic Manipulator

The capability of the multilayer perceptron for function approximation is further tested by training it via the preconditioned CGTR algorithm to approximate the inverse kinematics for a robotic manipulator. We briefly describe the inverse kinematics problem for a robotic manipulator. For further details the reader is referred to [23, 24, 25]. A mechanical manipulator or arm can be modeled as an interconnection of several links, each of which is connected to its predecessor through a *joint*. One end of the arm is attached to a base and the other end, to an *end-effector* or *gripper*. Robot manipulator kinematics deals with the analytic study of the motion of the robot arm with respect to a particular coordinate system. The *inverse kinematics* problem for a robotic manipulator involves the determination of the individual joint angles (angles between successive joints) $\underline{\theta}(t) = [\theta_1(t), \ldots, \theta_n(t)]^T$, given the spatial location of the end-effector $\underline{x}(t)$. This problem is solved using the equation

$$\underline{x}(t) = f(\underline{\theta}(t)), \tag{7}$$

where f is a nonlinear function. In general, for most manipulators there does not exist a continuous analytic f^{-1} over the whole space and even if it does, its solution can be analytically and numerically cumbersome.

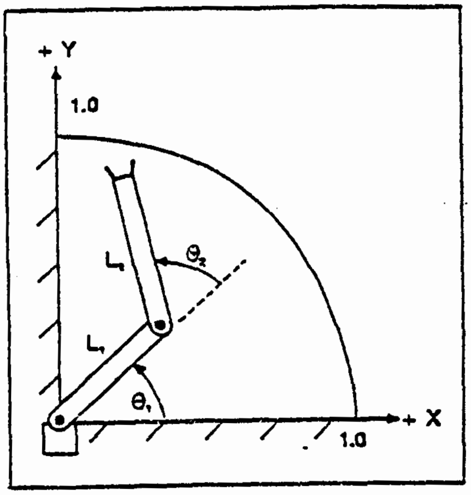


Figure 6: Bi-linked robot arm. Each link is assumed to be rigid and of length 0.5 units.

In this paper we attempt to solve the inverse kinematics problem by training the multilayer perceptron to approximate f^{-1}. The robotic manipulator arm considered in this study is a planar bi-linked arm, i.e., an arm with two segments constrained to lie in a single plane, as depicted in Fig 6. The joint angles $\theta_1(t)$ and $\theta_2(t)$ are calculated as

$$\begin{aligned} \theta_2(t) &= \cos^{-1}\left(\frac{x_1^2(t)+x_2^2(t)-L_1^2-L_2^2}{2L_1L_2}\right) \\ \theta_1(t) &= \tan^{-1}\left(\frac{x_2(t)}{x_1(t)}\right)-\tan^{-1}\left(\frac{L_2\sin\theta_2(t)}{L_1+L_2\cos\theta_2(t)}\right) \end{aligned} \quad (8)$$

The neural net is trained using the CGTR algorithm to approximate the above system of nonlinear equations.

Performance Analysis

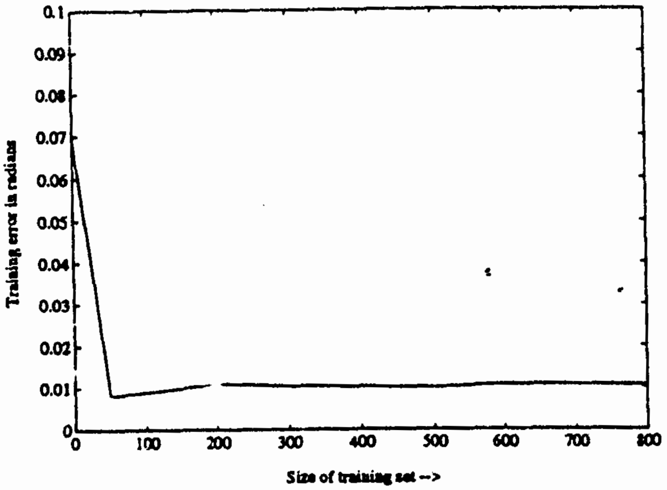


Figure 7: Multi-layer perceptron error function with increasing training set size.

The training is carried out by presenting pairs of input vectors and the corresponding desired joint angles to the neural net and minimizing the corresponding error function, as in (2), using the CGTR algorithm. The network configuration considered for this example is a three-layered network with 4 nodes in each of the middle layers. It has been observed that a three-layer network with as few as four nodes in each of the middle layers yields much better performance than a two-layer network with as many as 30 nodes in the middle layer. The number of input and output nodes correspond to the dimension of the work-space and the number of joint angles respectively. Figure 7 shows the change in the perceptron training error (for the three-layer perceptron with 4 nodes in each middle layer) with increasing training set size and as can be seen, the error stablizes after a certain point (400 points in this case).

V Conclusions

We have demonstrated the potential of the CGTR training algorithm for multilayer perceptrons and compared it to the existing backpropagation algorithm. The CGTR algorithm performed significantly better than backpropagation in the applications considered. Specifically, in the case of the detection of a single transmitter's signal in the presence of interfering users, the network trained with backpropagation was able to draw only a linear decision boundary compared to the near-optimum decision boundary obtained by training with the CGTR algorithm. Correspondingly, training with CGTR resulted in a lower probability of error. In the solution of the inverse kinematics problem, our precursory results have demonstrated the effectiveness of the CGTR algorithm in enabling the multilayer perceptron to successfully approximate the nonlinear functions involved. Further research is being carried out using two-layer perceptrons with a larger number of nodes in the middle layers as well as four-layr perceptrons.

References

[1] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, April, 1987.

[2] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representation by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations* (D. E. Rumelhart and J. L. McClelland, eds.), pp. 318–362, MIT Press, 1986.

[4] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Journal on Neural Networks*, vol. 2, pp. 359–366, 1989.

[5] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, Cambridge, MA, 1974.

[6] F. Rosenblatt, "Perceptions and the theory of brain mechanisms," in *Principles of Neurodynamics*, Washington, D.C: Spartan Books, 1962.

[7] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall Series in Computational Mathematics, Englewood Cliffs, New Jersey: Prentice-Hall Inc.,, 1983.

[8] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.

[9] M. Avriel, *Nonlinear Programming: Analysis and Methods*. Prentice-Hall Series in Automatic Computation, Englewood Cliffs, New Jersey: Prentice-Hall, 1976.

[10] T. Steihaug, "The Conjugate Gradient Method and Trust Regions in Large Scale Optimization," Tech. Rep. Rice MASC TR81-1, Department of Mathematical Sciences, Rice University, Houston, TX, 1982.

[11] T. Steihaug, *Quasi-Newton Methods for Large Scale Nonlinear Problems*. PhD thesis, Yale University, New Haven, Connecticut, 1980.

[12] H. V. Poor and S. Verdu, "Single-User Detectors for Multiuser Channels," *IEEE Trans. Commun.*, vol. COM-36, no. 1, pp. 50–60, January, 1988.

[13] S. Verdu, "Optimum Multiuser Asymptotic Efficiency," *IEEE Trans. Commun.*, vol. COM-34, pp. 890–897, September, 1986.

[14] B. Aazhang, B.-P. Paris, and G. Orsak, "Neural Networks for Multiuser Detection in Code-Division Multiple-Access Communications," *IEEE Trans. Commun.*, vol. COM-40, no. 7, July, 1992.

[15] J. J. More and D. C. Sorensen, "Computing a Trust Region Step," *SIAM J. SCI. STAT. COMPUT.*, vol. 4, pp. 553–572, September 1983.

[16] D. C. Sorensen, "Newton's Method with a Model Trust Region Modification," *SIAM J. Numer. Anal*, vol. 19, pp. 409–426, 1982.

[17] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Res. Nat. Bureau of Standards*, vol. 49, pp. 409–436, December 1952.

[18] M. R. Hestenes, *Conjugate Direction Methods in Optimization*. Applications of Mathematics, New York: Springer-Verlag, 1980.

[19] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore: The Johns Hopkins University Press, second ed., 1989.

[20] O. Axellson and V. A. Barker, *Finite Element Solution of Boundary Value Problems : Theory and Computation*. Orlando, Fla.: Academic Press, 1984.

[21] O. Axellson, "Solution of Linear Systems of Equations: Iterative Methods," in *Sparse Matrix Techniques* (V. Barker, ed.), vol. 572 of *Lecture Notes in Mathematics*, pp. 1–51, New York: Springer-Verlag, 1976.

[22] S. G. Nash, "Preconditioning of Truncated-Newton Methods," *SIAM J. SCI. STAT. COMPUT.*, vol. 6, pp. 599–616, July 1985.

[23] A. Deo, "Application of optimal damped least-squares method to inverse kinematics of robotic manipulators," Master's thesis, Rice University, 1991.

[24] R. G. K.S. Fu and C. Lee, *Robotics: Control, Sensing, Vision and Intelligence*. Industrial Engineering Series, McGraw Hill, 1987.

[25] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of Dynamic Systems, Measurement and Control*, pp. 163–171, September 1986.

ON PROBABILITY-POSSIBILITY TRANSFORMATIONS

George J. Klir
Department of Systems Science
Thomas J. Watson School of Engineering and Applied Science
State University of New York
Binghamton, New York 13902, U.S.A.

and

Behzad Parviz
Department of Mathematics and Computer Science
California State University
Los Angeles, California 90032, U.S.A.

ABSTRACT

Several probability-possibility transformations are compared in terms of the closeness of preserving second-order properties. The comparison is based on experimental results obtained by computer simulation. Two second-order properties are involved in this study: noninteraction of two distributions and projections of a joint distribution.

1. Introduction

During the last three decades or so, science has been undergoing a major paradigm shift involving attitudes towards uncertainty. The many facets of this paradigm shift are well described in a book by Smithson [1989].

The old paradigm is characterized by the pursuit of absoiutely certain knowledge or, if impossible, by resorting to probability theory, as the only legitimate mathematical tool to deal with the lack of certainty. The new paradigm, on the contrary, is not only tolerant of uncertainty, but it views uncertainty as an important resource in pursuing knowledge. While uncertainty is not desirable for its own sake, its role to counterbalance complexity is crucial when complexity is unmanageable or when dealing with it is prohibitively expensive [Zadeh, 1973]. When the solution to a problem is not required to be uncertainty-free, the computational complexity involved may often be substantially reduced [Traub, et al., 1983].

In order to utilize uncertainty as a strategic resource, we need to understand it as broadly as possible. It turns out that probability theory does not facilitate sufficiently broad framework for this purpose [Klir, 1989a]. As a recognition of the limitations of probability theory, two generalizations in mathematics have emerged. One of them is the generalization of classical set theory into fuzzy set theory, which allows us to deal with sets that do not have sharp boundaries [Zadeh, 1965; Klir and Folger, 1988]. The second is the generalization of classical measure theory into fuzzy measure theory, which allows us to deal with measures that are not additive [Sugeno, 1977; Wang and Klir, 1992]. These two theories can be combined.

Fuzzy set theory and fuzzy measure theory, as well as their combination, provide us with a very broad mathematical frameworks for investigating uncertainty, within which various special theories of uncertainty can be formulated. Only two of these special uncertainty theories are of our interest in this paper: classical probability theory and possibility theory. We assume elementary knowledge of these theories [Klir and Folger, 1988].

As we argue elsewhere [Klir and Parviz, 1992a], probability theory and possibility theory are complementary, but not comparable. They are capable of describing different types of uncertainty. It is often desirable to transform uncertainty described in one of the theories to the complementary description in the other theory [Dubois and Prade, 1986; Bharathi-Devi and Sarma, 1985; Leung, 1982; Moral, 1986; Klir, 1991]. Several distinct, transformations have been proposed in the literature for this purpose. Our aim in this paper is to compare these transformations in terms of the closeness of preserving second-order properties. The comparison is based on experimental results obtained by computer simulation.

The paper is a continuation of a previous study [Klir and Parviz, 1992b]. While the previous study focuses only on one second order property, joint distribution calculated from two noninteractive marginal distributions, this paper covers also projections of given joint distributions. Furthermore, it describes results based on a slightly different measure of uncertainty that the one employed in the previous study. The new measure of uncertainty emerged recently as a better justified alternative [Klir and Parviz, 1992c].

2. Probability-Possibility Transformations Investigated

In order to describe the probability-possibility transformations that are the subject of our experimental study, let

$$\mathbf{p} = (p_1, p_2, ..., p_n),$$

$$\mathbf{r} = (r_1, r_2, ..., r_n),$$

denote, respectively, an ordered probability distribution and the corresponding ordered possibility distribution. We assume that $p_i \geq p_{i+1}$ and $r_i \geq r_{i+1}$ for all $i = 1, 2, ..., n-1$. According to normalization requirements of the two theories, $p_1 + p_2 + ... + p_n = 1$ and $r_1 = 1$.

The first type of probability-possibility transformations $\mathbf{p} \leftrightarrow \mathbf{r}$ that are covered by our study are transformations based on ratio scales. They are expressed by the equations

$$r_i = \frac{p_i}{p_1}, \tag{1}$$

$$p_i = \frac{r_i}{r_1 + r_2 + ... + r_n} \tag{2}$$

The second type of transformations $\mathbf{p} \leftrightarrow \mathbf{r}$, which are often cited in the literature, were proposed by Dubois and Prade [1982, 1983, 1986]. They are defined by the equations

$$r_i = \sum_{j=1}^{n} \min(p_i, p_j), \tag{3}$$

$$p_i = \sum_{j=i}^{n} \frac{(r_j - r_{j+1})}{j}, \tag{4}$$

where $r_{n+1} = 0$ by convention.

The third type of transformations p ↔ r, which are asymmetric, were proposed by Dubois, Prade, and Sandri [1991]. In one direction, p → r, they are defined by the equation

$$r_i = \sum_{j=i}^{n} p_j. \tag{5}$$

In the other direction, r → p, they are defined by Eq. (4).

The fourth type of transformations p ↔ r, which were proposed by Klir [1989, 1990], are transformations that preserve uncertainty. It was shown by Geer and Klir [1992] that unique transformations of this kind exist only under log-interval scales. They are defined by Eqs. I and III in Figure 1. The value of α in these equations is determined by solving Eq. II in Fig. 1, which expresses the requirement that the amount of uncertainty be preserved when p is transformed to r or vice versa. Functions H, N, S in Eq. II are defined by the following formulas [Klir and Parviz, 1992c; Klir, 1993]:

$$H(p) = -\sum_{i=1}^{n} p_i \log_2 p_i, \tag{6}$$

$$N(r) = \sum_{i=2}^{n} (r_i - r_{i+1}) \log_2 i, \tag{7}$$

$$S(r) = \sum_{i=2}^{n} (r_i - r_{i+1}) \log_2 \frac{i}{\sum_{j=1}^{i} r_j}. \tag{8}$$

Function H is the well-known Shannon entropy [Klir and Folger, 1988], and functions N,S are referred to as nonspecificity and strife, respectively.

Since the value of S(r) is severely restricted when compared with the value of N(r), as shown by Geer and Klir [1991] and Klir and Parviz [1992c], the term S(r) plays a relatively minor role in Eq. II. To study the effect of this term on results, we performed experiments both with and without the term. Furthermore, we performed also experiments in which function S in Eq. II is replaced with function D defined by the form

$$D(r) = -\sum_{i=1}^{n-1} (r_i - r_{i+1}) \log_2 \left[1 - i \sum_{j=i+1}^{n} \frac{r_j}{(j-1)j} \right]. \quad (9)$$

This function, referred to as discord, was employed prior to the discovery of the latter justified function S (Klir and Parviz, 1992c]. We have performed experiments with both functions in order to compare their performance.

That is, our experiments involve six distinct probability-possibility transformations. The following are convenient abbreviations of these transformations:

- RC -- ratio-scale transformations defined by Eqs. (1) and (2);
- DP -- transformations proposed by Dubois and Prade [1983], which are defined by Eqs. (3) and (4);
- AS -- asymmetric transformations defined by Eqs. (4) and (5) [Dubois, Prade, and Sandri, 1991];
- NS -- transformations that preserve uncertainty, which are defined by Eqs. I - III in Fig. 1;
- N -- same as NS except that S(r) is excluded from Eq. II;
- ND -- same as NS except that S in Eq. II is replaced with function D defined by Eq. (9).

3. Description of Experiments

Two classes of simulation experiments regarding the six types of probability-possibility transformations are reported in this paper. The purpose of experiments of the first class is to compare the transformations by the estimated average degree to which they preserve joint distributions constructed from noninteractive marginal distributions. The estimates are obtained by experiments of two types.

In each experiment of the first type, marginal probability distribution $\mathbf{p}_1$ and $\mathbf{p}_2$ are chosen for some $n \geq 2$. Assuming that $\mathbf{p}_1$ and $\mathbf{p}_2$ are noninteractive, we calculate the joint probability distribution $\mathbf{p}$ by taking the pair-wise product of their components. Next, we convert $\mathbf{p}_1$, $\mathbf{p}_2$ into the corresponding marginal possibility distributions $\mathbf{r}_1$, $\mathbf{r}_2$ by each of the transformation methods and combine each pair of distributions by taking the pair-wise minimum of their components. This results in one joint possibility distribution for each transformation method, which we convert (using the same method) to the corresponding probability distribution $\mathbf{p}'$. Now, we determine the closeness of $\mathbf{p}'$ to $\mathbf{p}$ in terms of these criteria: Hamming distance, Euclidean distance, and the maximum error.

Experiments of the second type are similar, but they begin with given marginal possibility distributions, $\mathbf{r}_1$ and $\mathbf{r}_2$, for which the joint possibility distribution $\mathbf{r}$ is calculated by using the minimum operator. The given distributions are also transformed into the corresponding marginal probability distributions p_1 and p_2 by each of the method. Joint probability distribution p is then calculated for each pair p_1, p_2 and transformed (by the same method) to the corresponding possibility distribution r'. Finally, r and r' are compared in terms of the same criteria as in experiments of the first type.

The purpose of experiments of the second class is to compare the six transformations by the estimated average degree to which they preserve marginal distributions calculated from given joint distributions. Two types of experiments are again distinguished, depending on whether the inputs are probability distributions or possibility distributions. The results are compared in terms of the same criteria as in the experiments of the first class.

4. Experimental Results

Experiments of the two classes and both types were performed for selected values of n, from n=2 to n = 25. In each category and for each particular value of n, the performance of transformations RS, DP, and AS was compared with the three variants of uncertainty-preserving transformations, ND, NS, and N, in terms of the Hamming distance, the Euclidean distance, and the maximum error.

Results of experiments of the first class that are based on ND are published in a previous paper [Klir and Parviz, 1992b]. They demonstrate, with considerable consistency, that the uncertainty-preserving transformation performs best according to each of the three indicators and that its relative performance increases with increasing n. The results also show that AS is substantially outperformed by all the other transformations.

After these initial results, we extended the experiments of the first class to NS and N. We discovered that NS also performs better than RS, DP and AS, but it is slightly outperformed by ND. However, the difference between the two performances decreases with increasing n.

The behavior of N, which is illustrated by the selected data in Table I, is more interesting. While N is comparable with or even slightly weaker than RS and DP for small values of n (approximately n ≤ 5), it outperforms all the other transformations (including NS and ND) for large values of n (approximately n ≥ 10). For both types of experiments, the table is divided into three parts that contain values of the Hamming distance, the Euclidean distance, and the maximum error (in this order). Each column in the table represents one of the four conversion methods, as applied in experiments of either the first type or the second type. All entries in the table are average values based on 100 experiments for randomly selected marginal distributions.

Results of experiments of the second class for n = 5, 10, 15, 20, 25 are given in Table II (first type) and Table III (second type). As ir Table I, the three parts in either table contain values of the Hamming distance, the Euclidean distance, and the maximum error. Covered are all the six transformations introduced in Sec. 2.

We can see from Table III that each of the uncertainty-preserving transformations heavily outperforms transformations RS, DP, and AS in experiments of the second type. The order of the transformations by their performance is consistently N, NS, ND, DP, RS, AS according to each of the three indicators. The strong performance of transformations N is rather surprising.

According to experiments of the first type (Table II), the transformations are less discriminated by their performance, but NS (and ND, to a lesser degree) consistently outperforms the other transformations. The performances of N, RS, and DP are comparable and consistently higher than the performance of AS.

5. Conclusions

From all experimental results obtained in this experimental study, which are exemplified by five selected values of n in Tables I - III and in our previous paper [Klir and Parviz, 1992b], we may conclude that the uncertainty-preserving transformations are superior in terms of the degree to which they preserve the two second order properties investigated. This is not surprising since the uncertainty-preserving transformations neither loose information nor add extraneous information by the transformation process itself. It is reasonable to expect that the same conclusion will be obtained for other second order properties, such as conditioning or joining of overlapping distributions. We intend to validate this conjecture by additional experiments.

Although we consider three variants of uncertainty-preserving transformations, NS, ND and N, the differences among their performances are not large. This is a result of the fact that functions S and D are bounded from above by the same value, which is rather small [Geer and Klir, 1991]. One the whole, transformation N appears to be the best choice, not only due to its high performance in most cases, but also due to its simplicity.

Functions S and D (and the associated functions NS and ND) are still somewhat controversial as measures of possibilistic uncertainty, while function N alone does not represents the whole uncertainty [Klir and Parviz, 1992c; Klir, 1993]. If this controversy is resolved by determining a fully-justified measure of total uncertainty, the performance of the resulting uncertainty-preserving transformation will almost certainly outperform all the three currently considered uncertainty-preserving transformations.

Transformations NS, ND and N are based on log-interval scales and, as a consequence, they are unique. Uncertainty-preserving transformations may also be based on ordinal scales. Such transformations, which are not unique, may give us a greater flexibility in achieving desirable results, such as preserving best certain second order properties of the given distributions, maximizing the degree of probability-possibility consistency, and the like. A formulation of ordinal-scale transformations that preserve uncertainty and discussions of several other issues regarding probability-possibility transformations are included in another paper [Klir and Parviz, 1992a].

Acknowledgment

This work was supported in part by the National Science Foundation under Grant No. IRI-90 15675.

REFERENCES

Bharathi-Devi, B. and V.V.S. Sarma [1985], "Estimation of fuzzy memberships from histograms." *Information Sciences*, 35(1), pp. 35-59.

Delgado, M. and S. Moral [1987], "On the concept of possibility-probability consistency." *Fuzzy Sets and Systems*, 21(3), pp. 311-318.

Dubois, D. and H. Prade [1982], "On several representations of an uncertain body of evidence." In: *Fuzzy Information and Decision Processes*, ed. by M.M. Gupta and E. Sanchez, North-Holland, New York, pp. 167-181.

Dubois, D. and H. Prade [1983], "Unfair coins and necessity measures: towards a possibilistic interpretation of histograms." *Fuzzy Sets and Systems*, 10(1), pp. 15-20.

Dubois, D. and H. Prade [1986], "Fuzzy sets and statistical data." *European J. of Operations Research*, 25, pp. 345-356.

Dubois, D., H. Prade, and S. Sandri [1991], "On possibility/probability transformations." *Proc. Fourth IFSA Congress:Mathematics*, Brussels, pp. 50-53.

Geer, J.F. and G.J. Klir [1991], "Discord in possibility theory." *Intern. J. of General Systems*, 19(2), pp. 119-132.

Geer, J.F. and G.J. Klir [1992], "A mathematical analysis of information-preserving transformations between probabilistic and possibilistic formulations of uncertainty." *Intern. J. of General Systems*, 20(2), pp. 143-176.

Klir, G.J. [1989a], "Is there more to uncertainty than some probability theorists might have us believe?" *Intern. J. of General Systems*, 15(4), pp. 347-378.

Klir, G.J. [1989b], "Probability-possibility conversion." *Proc. 3rd IFSA Congress*, Seattle, pp. 408-411.

Klir, G.J. [1990], "A principle of uncertainty and information invariance." *Intern. J. of General Systems*, 17(2-3), pp. 249-275.

Klir, G.J. [1991], "Some applications of the principle of uncertainty invariance." *Proc. Intern. Fuzzy Eng. Symp.*, Yokohama, Japan.

Klir, G.J. [1993], "Developments in uncertainty-based information." In: *Advances in Computers*, Vol. 35, ed. by M.C. Yovits, Academic Press, San Diego.

Klir, G.J. and T.A. Folger [1988], *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliff, NJ.

Klir, G.J. and B. Parviz [1992a], "Probability-possibility transformations: A comparison." *Intern. J. of General Systems*, 21(4), pp.

Klir, G.J. and B. Parviz [1992b], "Possibility-probability conversions: an empirical study." In: *Progress in Cybernetics and Systems*, ed. by R. Trappl, Hemisphere, New York.

Klir, G.J. and B. Parviz [1992c], "A note on the measure of discord." *Proc. Eighth Conf. on Uncertainty in Artificial Intelligence*, Stanford.

Leung, Y. [1982], "Maximum entropy estimation with inexact information." In: *Fuzzy Set and Possibility Theory*, ed. by R.R. Yager, Pergamon Press, Oxford, pp. 32-37.

Moral, S. [1986], "Construction of a probability distribution from a fuzzy information." In: *Fuzzy Sets Theory and Applications*, ed. by A. Jones, A. Kaufmann, and H.J. Zimmermann. D. Reidel, Dordrecht, pp. 51-60.

Smithson, M. [1989], *Ignorance and Uncertainty: Emerging Paradigms*, Springer-Verlag, New York.

Sugeno, M. [1977], "Fuzzy measures and fuzzy integrals: a survey." In: *Fuzzy Automata and Decision Processes*, edited by M.M. Gupta, G.N. Saridis, and B.R. Gaines, North-Holland, Amsterdam and New York, pp. 89-102.

Traub, J.F., G.W. Wasilkowski, and H. Wozniakowski [1983], *Information, Uncertainty, Complexity*. Addison-Wesley, Reading, Mass.

Wang, Z. and G.J. Klir [1992], *Fuzzy Measure Theory*. Plenum Press, New York.

Zadeh, L.A. [1965], "Fuzzy sets." *Information and Control*, 8(3), pp. 338-353.

Zadeh, L.A. [1973], "Outline of a new approach to the analysis of complex system and decision processes." *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-3(1), pp. 28-44.

TABLE I. Experiments of the first class (from marginal distributions to joint distributions).

	First Type				Second Type			
n	N	RS	DP	AS	N	RS	DP	AS
5	0.1701	0.1725	0.1632	0.7416	1.8982	2.0760	2.7515	6.6134
10	0.1776	0.2010	0.1894	0.8238	5.1834	6.5029	7.7820	17.6543
15	0.1772	0.2044	0.1963	0.8502	13.0548	19.1767	21.0311	45.3405
20	0.1727	0.2041	0.1954	0.8667	20.4573	30.3087	32.8968	71.5128
25	0.1791	0.2133	0.2055	0.8706	31.4490	46.9769	50.7536	106.9599
5	0.0444	0.0437	0.0464	0.3325	0.4047	0.5132	0.5642	1.3115
10	0.0231	0.0259	0.0291	0.2419	0.6549	0.9117	0.9821	2.1100
15	0.0151	0.0178	0.0207	0.1907	1.0176	1.5935	1.6742	3.3802
20	0.0112	0.0134	0.0155	0.1569	1.2712	2.0014	2.1135	4.2590
25	0.0092	0.0113	0.0130	0.1367	1.5634	2.4779	2.6058	5.0989
5	0.0207	0.0198	0.0296	0.3127	0.1549	0.2073	0.1968	0.4212
10	0.0058	0.0071	0.0163	0.2197	0.1597	0.2272	0.2460	0.4258
15	0.0026	0.0035	0.0102	0.1656	0.1546	0.2390	0.2959	0.4283
20	0.0015	0.0020	0.0066	0.1311	0.1550	0.2419	0.3019	0.4278
25	0.0010	0.0014	0.0050	0.1113	0.1550	0.2445	0.3115	0.4245

TABLE II. Experiments of the second class and first type.

n	NS	ND	N	RS	DP	AS
5	0.2490	0.2606	0.2473	0.2651	0.2676	0.3495
10	0.2006	0.2096	0.2024	0.2126	0.2152	0.2591
15	0.1812	0.1814	0.1849	0.1826	0.1842	0.2148
20	0.1518	0.1528	0.1555	0.1537	0.1551	0.1766
25	0.1415	0.1409	0.1452	0.1416	0.1426	0.1578
5	0.1317	0.1377	0.1309	0.1399	0.1412	0.1854
10	0.0775	0.0805	0.0783	0.0815	0.0825	0.1005
15	0.0571	0.0573	0.0583	0.0577	0.0582	0.0684
20	0.0418	0.0421	0.0428	0.0423	0.0427	0.0485
25	0.0350	0.0349	0.0359	0.0351	0.0353	0.0391
5	0.0945	0.0971	0.0944	0.0984	0.0993	0.1339
10	0.0466	0.0480	0.0472	0.0485	0.0490	0.0619
15	0.0300	0.0301	0.0306	0.0303	0.0306	0.0366
20	0.0202	0.0206	0.0207	0.0207	0.0209	0.0236
25	0.0158	0.0157	0.0161	0.0158	0.0159	0.0178

TABLE III. Experiments of the second class and second type.

n	NS	ND	N	RS	DP	AS
5	0.4136	0.5870	0.1866	1.2693	0.7449	2.2727
10	0.5109	1.0152	0.2233	2.7772	1.3854	5.1137
15	0.4282	1.2133	0.2005	4.0189	1.7846	7.8077
20	0.3843	1.3729	0.1935	5.1132	2.0985	10.4588
25	0.3592	1.5083	0.1941	6.1502	2.3972	13.1046
5	0.2481	0.3472	0.1106	0.7224	0.4711	1.2184
10	0.2121	0.4065	0.0926	1.0451	0.6150	1.8660
15	0.1407	0.3824	0.0660	1.1987	0.6485	2.3112
20	0.1087	0.3686	0.0551	1.3022	0.6646	2.6762
25	0.0895	0.3570	0.0490	1.3872	0.6799	2.9965
5	0.1974	0.2711	0.0879	0.5250	0.3833	0.8383
10	0.1394	0.2528	0.0613	0.5695	0.4221	0.9159
15	0.0787	0.1994	0.0373	0.5468	0.3971	0.9413
20	0.0559	0.1730	0.0288	0.5275	0.3799	0.9528
25	0.0419	0.1506	0.0237	0.5080	0.3631	0.9622

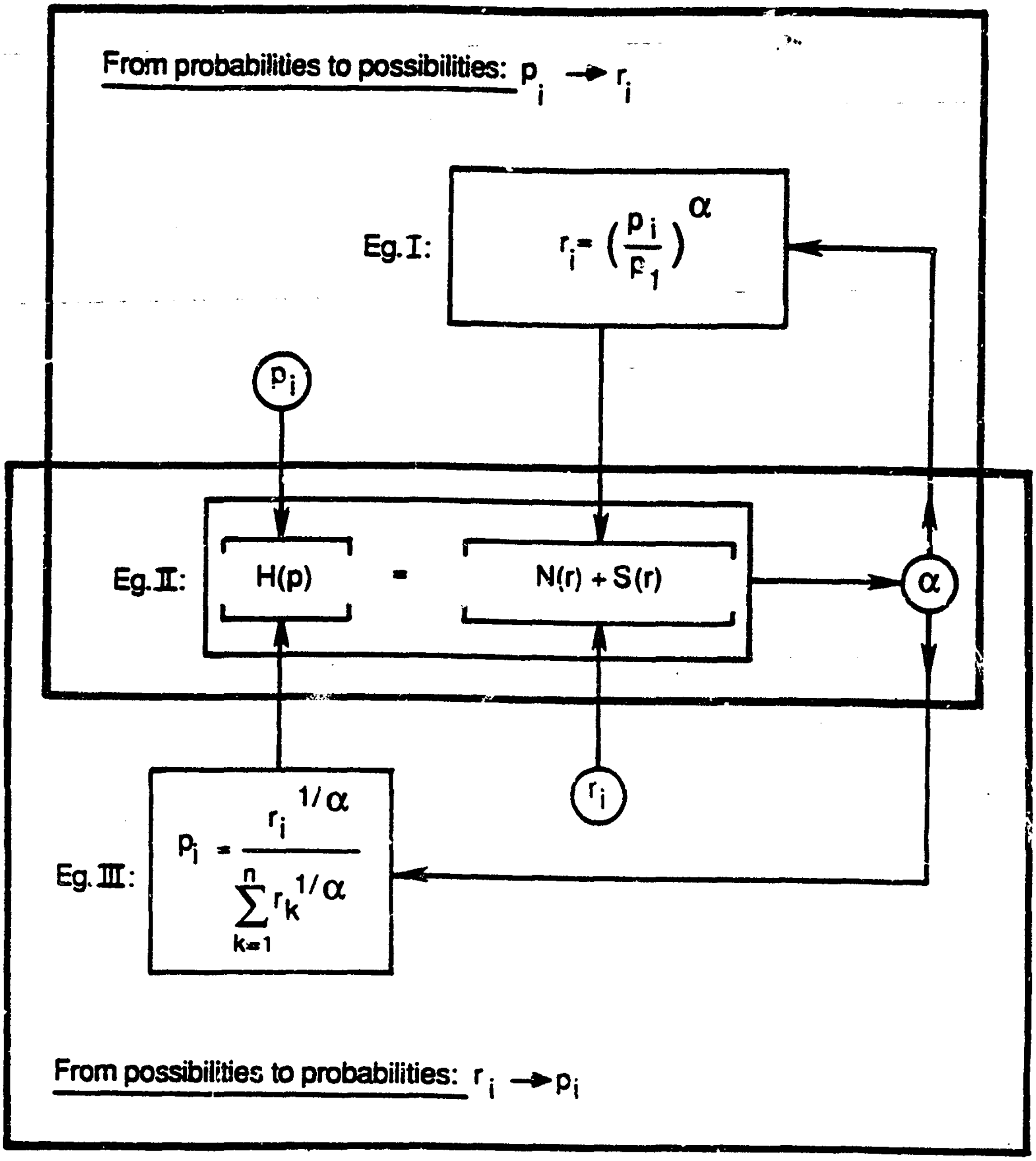

From probabilities to possibilities: $p_i \rightarrow r_i$
Eg. I:
$r_i = \left(\frac{p_i}{p_1}\right)^{\alpha}$
p_i
Eg. II:
H(p)
=
N(r) + S(r)
α
r_i
Eg. III:
$p_i = \frac{r_i^{1/\alpha}}{\sum_{k=1}^{n} r_k^{1/\alpha}}$
From possibilities to probabilities: $r_i \rightarrow p_i$

Figure 1. Uncertainty-invariant transformations between probabilities and possibilities based on log-interval scales.

Inference in fuzzy rule bases with conflicting evidence

László T. Kóczy[*]

Department of Telecommunication and Telematics,
Technical University of Budapest,
Sztoczek u. 2, Budapest H-1111, Hungary

1 Introduction

Inference based on fuzzy *'If ... then'* rules has played a very important role since when Zadeh [13] proposed the Compositional Rule of Inference and, especially, since the first succesful application presented by Mamdani *et al.* [10]. From the mid 1980's when the 'fuzzy boom' started in Japan, numerous industrial applications appeared, all using simplified techniques because of the high computational complexity. Another feature is that antecedents in the rules are distributed densely in the input space, so the conclusion can be calculated by some weighted combination of the consequents of the matching (fired) rules. The CRI works in the following way: If R is a rule and A^* is an observation, the conclusion is computed by $B^* = R \circ A^*$ ($\circ$ stands for the max-min composition). Algorithms implementing this idea directly have an exponential time complexity (maybe the problem is NP-hard) as the rules are relations in $X \times Y$, a $k_1 \times k_2$ dimensional space, if X is k_1, Y is k_2 dimensional. For a detailed analysis of the complexity see [3].

The simplified techniques usually decompose the relation into k_1 projections in X_i and measure in some way the degree of similarity between observation and antecedent by some parameter of the overlapping. These parameters are aggregated to a single value in $[0, 1]$ which is applied as a resulting weight for the given rule. The projections of rules in dimensions

[*] This work was done while a visiting appointment at the Department of Computer Science, Pohang Institute of Science and Technology, Pohang, Kyongbuk, P.O.Box 125, 790-330, Korea

Y_i are weighted by these aggregated values and then they are combined in order to obtain a resulting conclusion separately in every dimension.

This method is unapplicable with sparse bases as there is no guarantee that an arbitrary observation matches with any of the antecedents (cf. [14]). Then, the degree of similarity is 0 and all consequents are weighted by 0. Some considerations for such a situation are summarized in the next sections.

2 The semantical interpretation of inference

The rules we deal with in this paper have the form

'If X is A_i then Y is B_i'

Such a rule is represented by relation

$$R_i(x,y) = min\{A_i(x), B_i(y)\}$$

. This interpretes R_i as a 'fuzzy point' in $X \times Y$ and so the whole rule system describes in some way a fuzzy function $y = \mathcal{R}(x)$. For a thorough analysis of rule interpretations see [1].

An observation *'X is A^*'* is a fuzzy value of X and is transformed to $X \times Y$ in the form of its cylindric extension

$$O(x,y) = A^*(x)$$

. For rule system $R = \{R_i : i \in N_r\}$ the fuzzy conclusion in $X \times Y$ is

$$C(x,y) = max_i\{min\{R_i(x,y), O(x,y)\}$$

in $X \times Y$, and its projection to Y is

$$B^*(y) = sup_x\{max_i\{min\{R_i(x,y), O(x,y)\}$$

This algorithm of inference estimates the value $y = \mathcal{R}(A^*(x))$ by $B^*(y)$. $B^*(y) \neq 0$ only when the antecedent parts A_i cover the input space, i.e. for every x there is always at least one such rule R_i that $x \in supp(R_i)$. In sparse rule bases [14], this kind of inference results in no conclusion.

The approach of Türkşen dealing with this kind of problems [11] uses the similarity measure of two fuzzy sets: *similarity measure* = (1+*distance measure*)$^{-1}$. With the usual crisp distance measures of fuzzy sets, the similarity measure defined in this way has its range in [0,1], but it results in 0 when the two fuzzy sets have disjoint supports. In the next, this idea will be extended to arbitrary rule bases.

3 Gradual metric variables and fuzzy distance of fuzzy sets

Variables in real control applications have usually comparable and measurable values. In some other examples like when classifying tomatoes according to ripeness on the basis of their colours (see [4, 6]), a similar comparability and at least some 'pseudo-measurability' appears.

In [2] a very interesting interpretation of the semantic contents of fuzzy rules is proposed:

'If X is A then Y is B' = 'The more X is A the more Y is B'

The idea of gradual rules in [2] is in accordance with the analogical reasoning in [11] can be interpreted as:

'The more similar is x to A the more similar is y to B.

Gradual rules exist because the *variables* appearing in them are gradual. Graduality means mathematicaly that a full ordering can be defined over the variables. In practical cases, domain and range of the variables are finite, so $max\{X\}$, $min\{X\}$, etc. exist. If X and Y are compound, their components are bounded sets with a full ordering, so a partial ordering exists in both X and Y:

$$x_1 < x_2 \text{ iff } \forall i: \; x_{1,i} < x_{2,i} \text{ etc.}$$

Also the overall minima and maxima exist.

Beside ordering, measurability can be observed: as e.g., 100°C is farther from 12°C than from 67°C, etc. So the *distance* of two values can be expressed. In the case of many originally non measurable variables, some natural mapping of the range to the interval $[0, 1]$ provides virtual measurability. Variables measurable in any sense will be named *metric*. Even tomato colours or degrees of ripeness are metric so, as a mapping from $[deep\ green, deep\ red]$ to $[0, 1]$ can be introduced.

The fuzzy distance between linguistic (fuzzy) sets is defined with help of the Resolution Principle, for pairs of fuzzy sets satisfying the partial ordering $A \prec B$. $\prec$ is introduced over $\tilde{\mathcal{P}}(X_i)$, the set of all convex and normal fuzzy sets of X_i, so that for $A, B \in \tilde{\mathcal{P}}(X_i)$ $A \prec B$ if

$$\forall \alpha \in (0, 1]: \; inf\{A_\alpha\} < inf\{B_\alpha\} \text{ and } sup\{A_\alpha\} < sup\{B_\alpha\}$$

$\tilde{\mathcal{R}}_\prec$, a subset of $\tilde{\mathcal{P}}^2(X)$, is the relation of all comparable pairs:

$$\tilde{\mathcal{R}}_\prec = \{(A, B) | A, B \in \tilde{\mathcal{P}}(X), A \prec B\}$$

Given two fuzzy sets A and B in $\tilde{\mathcal{R}}_{\prec}(A,B)$, the *lower fuzzy* distance of A and B is

$$\tilde{d}_L(A,B) : \tilde{\mathcal{R}}_{\prec} \rightarrow \tilde{\mathcal{P}}([0,1]) \text{ and}$$

$$\mu_{\tilde{d}_L(A,B)}(\delta) = \sum_{\alpha \in [0,1]} \alpha / D(inf\{A_\alpha\}, inf\{B_\alpha\}),\ \delta \in [0,1] \text{ or } \delta \in [0,\sqrt{k_1}]$$

Similarly, the upper distance $\tilde{d}_U(A,B)$ can be defined. In the above, D stands for the Euclidean (or, more generally Minkowski) distance of A and B. (For more details see [9].)

Considering R from the point of view of the Resolution Principle, every rule is resolved to a family of α-rules:

'If X is $A_{i\alpha}$ then Y is $B_{i\alpha}$'

The α-cuts are represented by k_1- and k_2-dimensional hyperintervals in $X \times Y$. Every hyperinterval has its infimum and supremum, so if α is set fix, every rule can be unambiguously described by a *pair of points in* $X \times Y$ i.e. one for the infima ('lower point') and one for the suprema ('upper point'). With finite level sets of A_i and B_i it is sufficient to represent every rule by $2|\bigcup_i(\Lambda_{A_i} \cup \Lambda_{B_i})|$ points. In such a way every rule base consisting of r rules is represented in $X \times Y$ for given α and L or U by exactly r points.

4 Linear interpolation of rules

Extended gradual rules can be interpreted by the simple linear ratio:

$$dist(A^*, A_1) : dist(A^*, A_2) = dist(B^*, B_1) : dist(B^*, B_2)$$

$$\text{if } A_1 \prec A^* \prec A_2 \text{ and } B_1 \prec B_2$$

Interpreting *dist* as the fuzzy distance, the fundamental equation of linear rule interpolation is introduced:

$$\tilde{d}_\alpha(A_1, A^*) : \tilde{d}_\alpha(A^*, A_2) = \tilde{d}_\alpha(B_1, B^*) : \tilde{d}_\alpha(B^*, B_2)$$

$$\text{where } \alpha \in \Lambda_{A_1} \cup \Lambda_{A_2} \cup \Lambda_{B_1} \cup \Lambda_{B_2}$$

Applying the definition of $\tilde{d}_\alpha$ for L and U separately, altogether $2|\Lambda|$ equations are obtained. These can be solved:

$$inf_{\prec}\{B^*_\alpha\} = \frac{\frac{1}{\tilde{d}_{\alpha L}(A_{1,\alpha}, A^*_\alpha)} inf_{\prec}\{B_{1,\alpha}\} + \frac{1}{\tilde{d}_{\alpha L}(A^*_\alpha, A_{2,\alpha})} inf_{\prec}\{B_{2,\alpha}\}}{\frac{1}{\tilde{d}_{\alpha L}(A_{1,\alpha}, A^*_\alpha)} + \frac{1}{\tilde{d}_{\alpha L}(A^*_\alpha, A_{2,\alpha})}}$$

$$sup_{\prec}\{B^*_\alpha\} = \frac{\frac{1}{d_{\alpha U}(A_{1,\alpha},A^*_\alpha)} sup_{\prec}\{B_{1,\alpha}\} + \frac{1}{d_{\alpha U}(A^*_\alpha,A_{2,\alpha})} sup_{\prec}\{B_{2,\alpha}\}}{\frac{1}{d_{\alpha U}(A_{1,\alpha},A^*_\alpha)} + \frac{1}{d_{\alpha U}(A^*_\alpha,A_{2,\alpha})}}$$

So the α-level set of the conclusion is given for every α:

$$B^*_\alpha = [inf_{\prec}\{B^*\}, sup_{\prec}\{B^*\}]$$

and so the fuzzy set B^* can be constructed. Fig. 1 depicts a simple example for interpolating the coclusion belonging to a non-overlapping observation. On Fig. 2.a the α-distance (lower) for two comparable fuzzy sets is indicated, b shows the fuzzy distance sets.

It is possible to extend this idea to the interpolation of $2k$ rules, further on to various modified techniques of rule inter- and extrapolation. For more details, see [4, 5, 7, 8].

5 Approximation of the conclusion by regression

A very difficult question is what happens if the rule base contains some internal conflicts. An extremal example for this is if for any α and L or U there are two different rules in the base for which $min/maxA_{i1\alpha} = min/maxA_{i2\alpha}$ but $min/maxB_{i1\alpha} \neq min/maxB_{i2\alpha}$. Then, any 'interpolation' results into a 'perpendicular' line in $1+1$ dimensions and no defined extrapolation outside the two rules. Also, in the case of simply applying the interpolation technique for two flanking rules it is not clear, which of the two must be taken into consideration.

Although there is no formal strict contradiction still we face conflicting evidence where the hypothetical approximation curve (e.g. polynomial interpolation) has a too large 'amplitude' and the interpolated parts are very far from the area in $X \times Y$ where the actual rules are located. Fig. 3.a presents a case with 6 rules where the approximation curve (using the extension of the above interpolation technique) fits the rules very well. In b however, the curve is rather different from the obvious behaviour of the rules, it goes outside of the 'rule area' and is rather far from the expected $\mathcal{R}(x)$. In such cases, instead of eliminating conflicting rules the situation should be accepted, as it is and the solution should be looked for in the form of some compromise and simultaneous consideration of the conflicting rules.

How can conflicting rules be calculated with simultaneously? A possible technique is based on linear regression (see e.g. [12]). As the rule system is represented by a set of points in $X \times Y$ it is reasonable to compute the best fitting straight line by the least square method. In $1+1$ dimensions this is defined by

$$y = ax + b = \frac{\sum x_i y_i - \sum x_i \sum y_i/r}{\sum x_i^2 - \sum x_i^2/r} x + (\sum y_i/r - a \sum x_i/r)$$

It is much more complicated to treat compound variables. If X has k_1 and Y has k_2 components, the least square regression will result a $k_1 \times k_2$ dimensional hyperplane. The problem can be always decomposed into k_2 $k_1 + 1$-dimensional problems where Y_i is approximated by $\sum_{j=1}^{k_1} a_{ij} x_j + b_i$. So it is sufficient to examine the case with compound X but simple Y.

The solution of this problem is given by

$$a = [a_i] \text{ and } a = \sum_i y_i/r - a^T [\sum_j x_{ij}] \text{ where}$$

$$a = ([x_{ij} - \sum_j x_{ij}/r]^T [x_{i,} - \sum_j x_{ij}/r])^{-1} [x_{ij} - \sum_j x_{ij}/r]^T [y_i - \sum_i y_i/r]$$

$i = 1...r$, $j = 1...k_1$, [ ] stands for indicating a matrix, ^{T} is the transposed.

It is clear that this regression line or hyperplane gives a very rough approximation of the rule base except if it has a really linear tendency. (See e.g. the rule base on Fig. 4.) So it is more reasonable to calculate $y = ax + b$ only for a given environment of the observation: a 'window' around the respective value of A^*. Then, we obtain the best fitting straight line only for a restricted area. If the window is not too large, this leads to a rather good partially linear approximation. (See Fig. 5 for the same rule base.)

Let us compare the window-regression technique with the previous interpolation/extrapolation method. While in the latter it is sufficient to calculate the approximation curve (maybe partially linear) once before starting the inference/control algorithm based on the rules, it seems that because of shifting the window it is necessary to calculate a new equation for y every time when we have a new observation. This is painful, especially in the compound case as the matrix inversion takes a very long runtime. If this is true, the computational complexity of the newly proposed method is not competitive with other techniques. Luckily enough, *in the case of simple variables* *and a rule base with r rules it is sufficient to calculate maximally $2r$ different regression lines for any fixed set of points and even in the compound variable case the space $X \times Y_i$ can be divided into maximally $2r^{k_1}$ areas where the regression hyperplane would be different.* (α and lower/upper). Proof of this statement is not very difficult. Fig. 6 depicts a simple example how to divide X, for the rule base of the previous figures. A consequence is that when using trapezoidal rules and $k_1 + k_2$ variables, altogether $8k_2r_1^k$ regression hyperplanes are necessary before starting a real time control. So it is guaranteed that computational time during the actual control is not higher than in the case of straightforward approximation.

Significant disadvantage of this technique is that the function obtained is not continuous: it is a broken line or broken plane and so the approximated conclusion might change abruptly when the observation is only slightly different. (See Fig. 7 as illustration to this.) A solution of this problem is presented by the application of the *fuzzy window technique*, i.e. the above method is modified so that the environment of every observation has fuzzy rather than crisp boundaries. So the abrupt appearance of a new rule in the window when the observation is moved slightly is eliminated completely: every rule appearing in the window is weighted by the membership value attached to the location of that rule - depending on the location of the observation - this weight is however very small if the window is defined by a membership function smooth enough. For this purpose, a trapezoidal window is rather suitable. (See Fig. 8. The areas with $\mu = 0$ and 1 are indicated, in between, $0 < \mu < 1$.)

It is a new problem now, how the least square method works with weighted points. Clearly, the gain in the smoothness and continuity of the approximation function costs considerable computational time. Because of the introduction of the fuzzy (continuous membership function) window, no equivalent or extension of the above statement concerning the finiteness of the number of possible regression lines exists. The regression line calculated in terms of the observation is continuously changing. An exact examination of the computational complexity will follow.

Instead of examining just the case of the fuzzy window regression we present the solution of the general fuzzy regression, where points can be weighted by arbitrary membership degrees.

Suppose that we have points (x_i, y_i) $(i = 1...r)$ and each has the membership degree μ_i. *The straight line with least square sum of difference is then*

$$y = ax + b = \frac{\sum \mu_i x_i (y_i - \frac{\sum \mu_i y_i}{\sum \mu_i})}{\sum \mu_i x_i (x_i - \frac{\sum \mu_i x_i}{\sum \mu_i})} x + (\frac{\sum \mu_i y_i}{\sum \mu_i} - \frac{\sum \mu_i x_i}{\sum \mu_i} a)$$

Proof of this statement is by partial differentiation of the residual sum of squares according to a and b.

Using the above, it is possible to approximate α L or U points of the conclusion in a highly flexible way: even flexible windows can be applied – as a matter of course with the computational time following from the above equations. It is not difficult to extend the above result for compound variable cases. Instead of giving the rather complicated equation we just indicate that the mean values $\sum_j x_{ij}/r$ and $\sum_i y_i/r$ must be replaced by $\sum_j \mu_j x_{ij} / \sum_i \mu_i$ and $\sum_i \mu_i y_i / \sum_i \mu_i$, resp. further on, in all the sums x_i is replaced by $\mu_i x_i$. It is a rather serious problem here that computational complexity is high, in every step of inference the inversion of several $r \times k_1$ dimensional matrices is to be done – depending on the cardinality of level sets at least 3 or 4 of them.

A further direction of this research is that σ^2 be calculated inside the crisp window, moreover, by using μ_i for weighting, fuzzy variance is obtained which can be used for measuring the degree of conflict in the evidence of the given rule base – on level α and L or U.

References

[1] D. Dubois and H. Prade: Basic issues on fuzzy rules and their application to fuzzy control. *Fuzzy Control Workshop, IJCAI-91*, Sydney, 1991. 13 p.

[2] D. Dubois and H. Prade: Gradual inference rules in approximate reasoning. To appear in *Information Sciences*, 1992.

[3] L. T. Kóczy: On the computational complexity of rule base fuzzy inference. *NAFIPS - '91*, Columbia, Missouri, 1991. pp. 87-91.

[4] L. T. Kóczy and K. Hirota: Rule interpolation in approximate reasoning based fuzzy control. *Proc. of Fourth IFSA World Congress*, Brussels, 89-92 (1991)

[5] L. T. Kóczy and K. Hirota: Rule interpolation by α-level sets in fuzzy approximate reasoning. *BUSEFAL 46* 115-123 (1991)

[6] L. T. Kóczy and Á. Juhász: Fuzzy rule interpolation and the RULEINT program. *Proc. Joint Hungarian – Japanese Symposium on Fuzzy Systems and Applications*, Budapest, 91-94 (1991)

[7] L. T. Kóczy, K. Hirota and Á. Juhász: Interpolation of 2 and $2k$ rules in fuzzy reasoning. *Proc. of IFES '92, Fuzzy Engineering toward Human Friendly Systems I.*, Yokohama, 206-217 (1991)

[8] L. T. Kóczy and K. Hirota: Reasoning by analogy with fuzzy rules. *Proc. IEEE Int. Conference on Fuzzy Systems*, San Diego, California 263-270 (1992)

[9] L. T. Kóczy and K. Hirota: Interpolative reasoning with insufficient fuzzy rule bases. *Submitted to Information Sciences* (1992)

[10] E. H. Mamdani and S. Assilian: An experiment in linguistic synthesis with a fuzzy logic controller. In: *E. H. Mamdani and B. Gaines (eds.): Fuzzy reasoning and its applications.* Academic Press, London, 1981. pp. 311-323

[11] I. B. Türkşen and Z. Zhong: An approximate analogical reasoning approach based on similarity measures. *IEEE Transactions on Systems, Man and Cybernetics* 1049-1056 (1988).

[12] S. Weisberg: Applied linear regression. Wiley, New York, 1980. 283p.

[13] L. A. Zadeh: Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems, Man and Cybernetics (1973).* pp. 28-44.

[14] L. A. Zadeh: Interpolative reasoning in fuzzy logic and neural network theory. *IEEE International Conference on Fuzzy Systems*, San Diego, California, 1992.

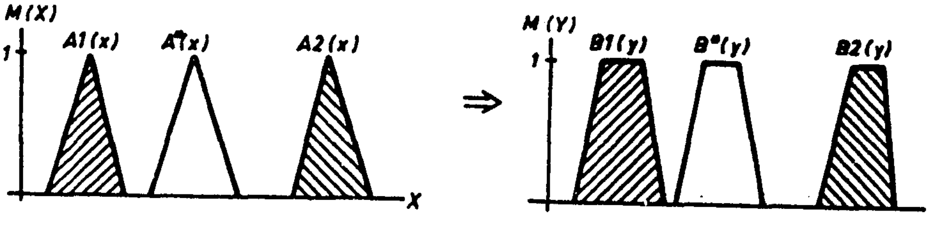


Fig. 1

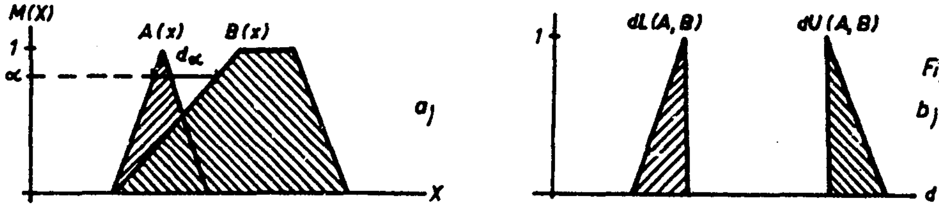


Fig. 2

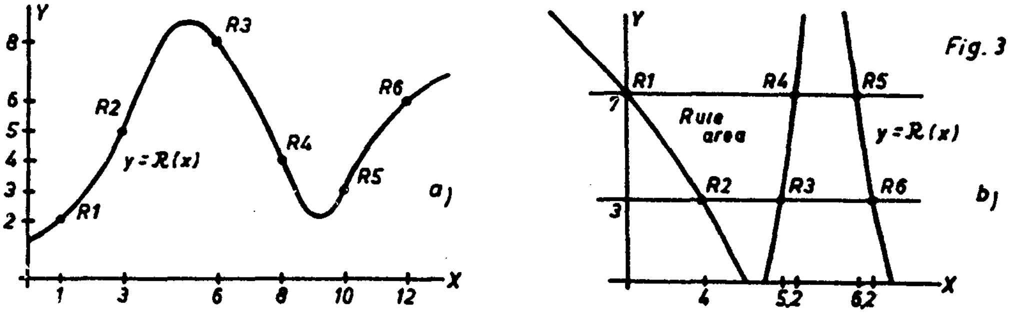


Fig. 3

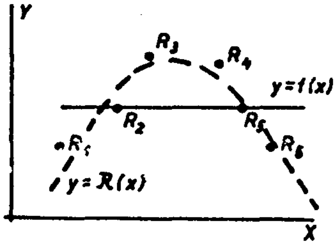


Fig. 4

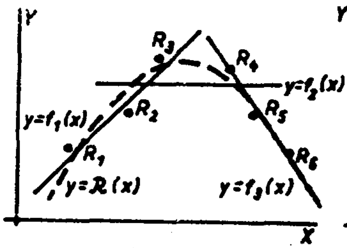


Fig. 5

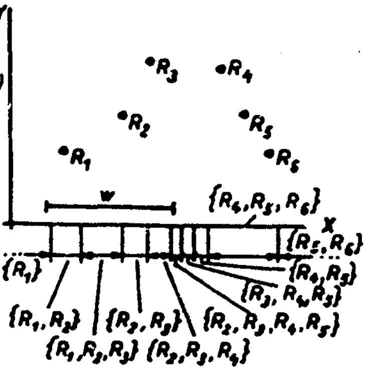


Fig. 6

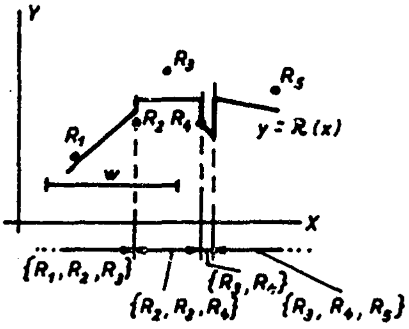


Fig. 7

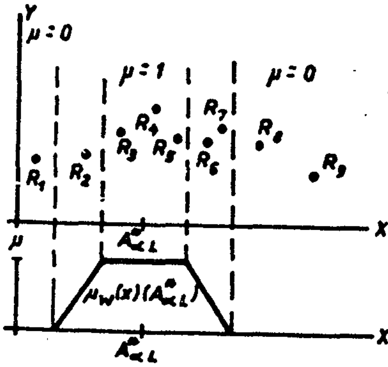


Fig. 8

N93

29584

UNCLAS

N93-29~~584~~585
161868
p- 8

GAUSSIAN MEMBERSHIP FUNCTIONS ARE MOST ADEQUATE IN REPRESENTING UNCERTAINTY IN MEASUREMENTS

V. Kreinovich[1], C. Quintana[1,2], L. Reznik[3]
[1]Computer Science Department, University of Texas at El Paso, El Paso, TX 79968 USA
[2]Department of Electrical Engineering and Computer Science,
University of Michigan at Ann Arbor, Ann Arbor, MI 48109-2122, USA
[3]Department of Electrical and Electronic Engineering, Footscary Campus,
Victoria University of Technology, MMC Melbourne, VIC 3000, Australia

Abstract. In rare situations like fundamental physics we perform experiments without knowing what their results will be. In the majority of real-life measurement situations, we more or less know beforehand what kind of results we will get. Of course, this is not the precise knowledge of the type "the result will be between $a - \delta$ and $a + \delta$", because in this case, we would not need any measurements at all. This is usually a knowledge that is best represented in uncertain terms, like "perhaps (or "most likely", etc.) the measured value x is between $a - \delta$ and $a + \delta$".

Traditional statistical methods neglect this additional knowledge and process only the measurement results. So it is desirable to be able to process this uncertain knowledge as well. A natural way to process it is by using fuzzy logic. But there is a problem: we can use different membership functions to represent the same uncertain statements, and different functions lead to different results. What membership function to choose?

In the present paper, we show that under some reasonable assumptions, Gaussian functions $\mu(x) = exv(-\beta x^2)$ are the most adequate choice of the membership functions for representing uncertainty in measurements. This representation was efficiently used in testing jet engines for airplanes and spaceships.

1. INTRODUCTION

Usually in measurement situations there is some prior knowledge. In rare situations like fundamental physics we perform experiments without knowing what their results will be. In the majority of real-life measurement situations, we more or less know beforehand what kind of results we will get. Of course, this is not the precise knowledge of the type "the result will be between $a - \delta$ and $a + \delta$", because in this case we would not need any measurements at all. This is usually a knowledge that is best represented in uncertain terms, like "perhaps (or "most likely", etc.) the measured value x is between $a - \delta$ and $a + \delta$".

Traditionally the uncertain prior knowledge is not used in measurement processing. Traditional statistical methods neglect this additional knowledge and process only the measurement results. So it is desirable to be able to process this uncertain knowledge as well.

The usage of fuzzy logic and related problems. A natural way to process uncertainty is by using fuzzy logic [Z65]. This way we represent every statement of the type "most likely, $|x - a| \leq \delta$" by a membership function $\mu(x)$ that for each x gives us a degree to which we are certain that this particular x satisfies the given condition. But there is a problem: we can use different membership functions to represent the same uncertain statements. What membership function to choose?

What we are planning to do? In the present paper, we show that under some reasonable assumptions, Gaussian functions $\mu(x) = exp(-\beta x^2)$ are the most adequate choice of the membership functions for representing uncertainty in measurements. This representation was efficiently used in testing jet engines for airplanes and spaceships.

2. MOTIVATION OF THE FOLLOWING DEFINITIONS

We must have in mind that different experts can have different opinions. Therefore the final resulting knowledge about the value of a physical quantity does not consist of a single statement, but can be formed by adding several statements of several expert, e.g., "most likely, $|x-a_1| \leq \delta_1$", "most likely, $|x-a_2| \leq \delta_2$", ... The resulting statement is "most likely, $|x-a_1| \leq \delta_1$, and most likely, $|x-a_2| \leq \delta_2$,..." In order to represent this resulting knowledge we must choose some operation $*$ for &. Then the resulting membership function will be equal to $\mu(x) = \mu_1(x)*\mu_2(x)*...$, where $\mu_i(x)$ corresponds to the opinion of i-th expert.

What &-operation to choose? Experimental results given in [HC76], [O77], and [Z78], show that among all possible operations $a, b \rightarrow min(a,b)$ and $a, b \rightarrow ab$ are the best fit for human reasoning.

The *min* operation does not seem to be adequate for our purposes, because if we use *min*, then, e.g., the degree, to which a function $x(t)$ satisfies the condition "for all t, most likely $|x(t)| \leq M$", is equal to the minimal of the degrees of the statements "most likely, $|x(t)| \leq M$" for all t. This minimum is attained when the value of $|x(t)|$ is the biggest possible. Therefore, the function $x_1(t)$ that is everywhere equal to $2M$, gets the same degree of consistency with the above-given rule, as the function that is almost everywhere equal to 0, and is attaining the value $2M$ only on a small interval. Intuitively, however, for the first function, for which the inequality is not true in a single point, our degree of belief that $x_1(t)$ satisfies this condition is practically 0, while for the second function, for which this inequality is almost everywhere true, our degree of belief must be close to 1.

So, using *min* in our problem is inconsistent with our intuition, and therefore we must use the product for &.

Comment. Other arguments for choosing different & operations are given in our previous publications [KR86] and [KQLFLKBR92].

We want to describe membership functions for the following statements. We are interested in describing statements of the type "most likely, $|x-a| \leq \delta$", where x is unknown, and a, δ are known values. So we must describe, to what extent any given value x satisfies this condition.

All these membership functions can be obtained from one of them. Evidently, x satisfies the inequality $|x-a| \leq \delta$ if and only if the value $y = (x-a)/\delta$ satisfies the inequality $|y| \leq 1$. Therefore, it is natural to assume that the statement "most likely, $|x-a| \leq \delta$" has the same degree of belief as the statement "most likely, $|y| \leq 1$", where $y = (x-a)/\delta$. So, if we will be able to describe a membership function $\mu(y)$ that corresponds to the statement "most likely, $|y| \leq 1$", then we will be able to describe our degree of belief $\mu_1(x)$ that x satisfies the condition "most likely, $|x-a| \leq \delta$" as $\mu((x-a)/\delta)$. So the main problem is to find an appropriate function $\mu(x)$.

What if we ask several experts. A statement "most likely, $|x-a| \leq \delta$" means that an expert estimates x as a, and his own estimate of his precision is $\approx \delta$. Since such estimates are often very crude, it is reasonable to ask the opinion of several experts. After we have asked k experts, we get k statements of the same type: "most likely, $|x-a_i| \leq \delta_i$", where $i = 1, 2, ..., k$, and a_i and δ_i are the estimates of the i-th expert. The corresponding membership functions are $\mu((x-a_i)/\delta_i)$.

Since all of them are experts, we believe in what all of them say, and therefore our resulting knowledge is: "most likely, $|x-a_1| \leq \delta_1$, and most likely, $|x-a_2| \leq \delta_2$, and ..." Since we agreed to represent "and" as a product, the resulting membership function is equal to $\nu(x) = \mu((x-a_1)/\delta_1)\mu((x-a_2)/\delta_2)...\mu((x-a_k)/\delta_k)$.

In case we have a precise knowledge, and each of the experts describes an interval, in which the unknown value x must be, the resulting knowledge is that x belongs to the intersection of all these intervals. This intersection is itself an interval, and therefore the only effect of asking several experts is that we decrease uncertainty. We do not change the form of the knowledge: it is still an interval, and in principle one smart expert could have named it from the very beginning.

In a similar way, it seems reasonable to assume that in the general fuzzy case, by combining the opinions of several experts, we do not seriously add any additional knowledge; we may diminish slightly an uncertainty domain for the unknown x, but that's all.

How to describe this argument mathematically: we must apply normalization. In mathematical terms, we would like to postulate that the resulting membership function $\nu(x)$ coincides with one of the functions $\mu((x-a)\delta)$, and in principle it could represent the opinion of just one smart expert.

We cannot, however, postulate precisely that. The reason is as follows. The bigger y, the smaller is our belief that "most likely, $|y| \leq 1$". So, the function $\mu(y)$ must be monotonously decreasing for $y > 0$. Its maximum m is attained, when $y = 0$. So, when we combine the two statements "most likely, $|x| \leq 1$", and "most likely, $|x - 0.3| \leq 1$", the resulting membership function $\nu(x) = \mu(x)\mu(x - 0.3)$ is always smaller than m^2, because both factors are $\leq m$, and for $x \neq 0$ the first factor is $< m$, and for $x = 0$ the second. So even if $m = 1$, the function $\nu(x)$ never attains m, and thus it cannot be equal to $\mu((x-a)/\delta$.

The solution to this problem is well known in fuzzy logic: we can *normalize* $\nu(x)$, i.e., turn from $\nu(x)$ to $\nu'(x) = N\nu(x)$, where the normalization constant N is equal to $N = 1/(max_y \nu(y))$.

Comment. A motivation for using namely this type of normalization is given in [KQLFLKBR92].

Now we are ready to formulate our demand.

3. MATHEMATICAL FORMULATION OF THE PROBLEM AND THE MAIN RESULT

Definition 1. By a *membership function* we will understand a continuous function $\mu(x)$ from the set R of all real numbers into the interval [0,1].

Definition 2. We say that two membership functions $\mu(x)$ and $\nu(x)$ are *equivalent* if $\mu(x) = C\nu(x)$ for some constant $C > 0$.

Definition 3. We say that a membership function $\mu(x)$ is *adequate for describing uncertainty of measurements* if it satisfies the following conditions:

- it is symmetric ($\mu(-x) = \mu(x)$),
- $\mu(x)$ is strictly decreasing on $(0, \infty)$ and tends to 0 as $x \to \infty$
- for every finite sequence of pairs $(a_1, \delta_1), (a_2, \delta_2), ..., (a_k, \delta_k)$ there exist a and δ such that the product $\mu((x - a_1)/\delta_1)\mu((x - a_2)/\delta_2)...\mu((x - a_k)/\delta_k)$ is equivalent to $\mu((x - a)/\delta)$.

THEOREM. *Any membership function, that is adequate for describing uncertainty of measurements, is equivalent to $exp(-\beta x)$ for some $\beta > 0$.*

(The proof is given in Section 5).

Comment. So we conclude that Gaussian functions are the only adequate membership functions. These functions are really widely used [K75], [BCDMMM85], [YIS85], [KM87, Ch. 5], etc. Alternative explanation of why Gaussian functions are used is given in [KR86] and in Section 8 of [KQLFLKBR92].

4. HOW THIS RESULT CAN BE USED AND HOW IT WAS USED

How it can be used. If for some physical quantity x several experts give their estimates a_1, a_2, ..., a_k, and they estimate the precision of their estimates as correspondingly $\delta_1, \delta_2, ..., \delta_k$, then the resulting membership function is equal to $\mu(x) = exp(-\beta(x-a)^2/\delta^2)$, where $\delta = (\delta_1^{-2} + \delta_2^{-2} + ... + \delta_k^{-2})^{-1/2}$ and $a = (a_1\delta_1^{-2} + ... + a_k\delta_k^{-2})/(\delta_1^{-2} + ... + \delta_k^{-2})$.

Comments.

1. These formulas can be easily obtained by explicitly computing $\mu(x)$ as a result of normalization of the product $\mu_1(x)\mu_2(x)...\mu_k(x)$, where $\mu_i(x) = exp(-\beta(x-a_i)^2/\delta_i^2)$.
2. These formulas are surprisingly identical with the statistical formulas that correspond to the case when we have k statistical estimates a_i with precisions δ_i and apply the least squares method $\sum_i (a-a_i)^2/\delta_i^2 \to max_a$ to get the resulting estimate for a. This is not such a big surprise, because least squares method is based on the assumption of a Gaussian distribution. The positive side is that not only the resulting formulas are extremely simple to implement, but maybe there is no need to implement them at all, because we can copy the existing statistical software.

How this result was actually used. Expert estimates are extremely important in testing the jet engines. The reason is that an important part of this testing is trying to figure out what is going on in the high-temperature regions, and the temperatures are so high there that we cannot place any sensors. So the only available information about these regions consists of the experts' estimates.

One of the authors (L.R.) used this fuzzy representation of uncertainty in designing software for the automatized jet engines testing system IVK-12 [KR86]. This system was actually used to test jet engine for aircraft and spaceships.

Possible other applications. One area where we believe this approach can be useful is when we determine the position of a Space Shuttle. The existing systems use several different types of sensors, with different precisions, and often with only experts estimates of that precision. In order to make appropriate control decisions we must combine these estimates into a single value. Fuzzy approach allows us to do that.

5. PROOF OF THE THEOREM

Comment. This proof contains some mathematical ideas from our previous publications [KR86] and [KQLFLKBR92].

1. Assume that $\mu(x)$ is an adequate function in the sense of the above definition. It is easy to check that if $\mu(x)$ is an adequate choice, then the result $\mu(x)/(max\, \mu(y))$ of its normalization is also an adequate choice. Since $\mu(x)$ is monotone, this maximum is attained for $x = 0$, and therefore the result of this normalization satisfied the condition $\mu(0) = 1$.

So, without losing any generality, we will further assume that $\mu(0) = 1$.

2. From the definition of an adequate function it follows, in particular, that $\mu(x)\mu(x) = C\mu((x-a)/\delta)$ for some a, C and δ. The left hand side attains its maximum ($= 1$) at $x = 0$, the right-hand side attains its maximum (that is equal to C) for $x = a$. Since these two sides are one and the same function, we conclude that $a = 0$ and $C = 1$, i.e., that $\mu^2(x) = \mu(k_2x)$ for some constant k_2 ($= 1/\delta$). For $l(x) = log\, \mu(x)$ we conclude that $2l(x) = l(k_2x)$.

Likewise, if we consider 3, 4, etc terms, we conclude that $3l(x) = l(k_2x)$, $4l(x) = l(k_4x)$, etc.

3. The function $\mu(x)$ for $x > 0$ is monotonously decreasing from 1 to 0. Therefore, $l(x)$ is monotonously decreasing from 0 to $-\infty$. Since μ is continuous, the function $l(x)$ is also continuous, and, therefore, there exists an inverse function $i(x) = l^{-1}(x)$, i.e., such a function that $i(l(x)) = x$ for every x.

For this inverse function, the equality $nl(x) = l(k_n x)$ turns into $i(nl(x)) = i(l(k_n x)) = k_n x = k_n i(l(x))$. So, if we denote $l(x)$ by X, we conclude that for every n, there exists a k_n such that $i(nX) = k_n i(X)$.

If we substitute $Y = nX$, we conclude that $i(Y) = k_n i(Y/n)$, and therefore, $i(Y/n) = (1/k_n)i(Y)$.

From these two equalities, we conclude that $i((m/n)X) = (1/k_n)i(nX) = (k_m/k_n)i(X)$. So, for every rational number r, there exists a real number $k(r)$ such that $i(rX) = k(r)i(X)$.

Therefore, the ratio $i(rX)/i(X)$ is constant for all rational r.

4. Since $i(X)$ is a continuous function, and any real number can be represented as a limit of a sequence of rational numbers, we conclude that this ratio is constant for real values of r as well. Therefore, for every real number r there exists a $k(r)$ such that $i(rX) = k(r)i(X)$.

All monotone solutions of this functional equation are known: they are $i(X) = AX^p$ for some A and p [A66]. Therefore, the inverse function $l(x)$ $(x > 0)$ also takes the similar form $l(x) = Bx^m$ for some k and m. Taking into consideration that $\mu(x)$ and hence $l(x)$ are even functions, we conclude that $l(x) = B|x|^m$ for all x.

5. Now, from the demand that a function $\mu(x)$ is adequate, we conclude that for every $a > 0$ we have $\mu(x-a)\mu(x+a) = C\mu((x-a_1)/\delta)$ for some a_1 and δ. The left-hand side of this equation is an even function, so the right-hand side must also be even, and therefore $a_1 = 0$. So, $\mu(x-a)\mu(x+a) = C\mu(x/\delta)$. For $x = 0$ we get $\mu(a)\mu(a) = C$. Turning to logarithms, we conclude that for every a, there exists a $k(a)$ such that $l(x-a)+l(x+a) = l(k(a)x)+2l(a)$. If we substitute here $l(x) = B|x|^m$, and divide both sides by B, we conclude that $|x-a|^m + |x+a|^m = k(a)^m|x|^m + 2a^m$.

6. When $x > 0$, and a is sufficiently small, then $x + a$, x, and $x - a$ are all positive, and, therefore, $(x-a)^m + (x+a)^m = k(a)^m x^m + 2a^m$. If we move $2a^m$ to the left-hand side, and divide both sides by x^m, we conclude that $(1-(a/x))^m + (1+(a/x))^m - 2(a/x)^m = k(a)^m$. The left-hand side of the resulting equality depends only on $z = a/x$, the right-hand side only on a. Therefore, if we choose any positive real number λ, and take $a' = \lambda a$ and $x' = \lambda x$ instead of a and x, then we can conclude that the left-hand side will be still the same, and therefore, the right-hand side must be the same, i.e., $k(a)^m = k(\lambda a)^m$. Since λ was an arbitrary number, we conclude that $k(a)$ does not depend on a at all, i.e., $k(a)^m$ is a constant. Let us denote this constant by k.

So the equation takes the form $(1-z)^m + (1+z)^m = k + 2z^m$. When $z \to 0$, then the left-hand side tends to 2 and right-hand side to k, so from their equality we conclude that $k = 2$.

The left-hand side is an analytical function of z for z close to 0. Therefore the right-hand side must also be a regular analytical function in the neighborhood of 0 (i.e., it must have a Taylor expansion for $z = 0$). Hence, m must be an integer.

The values $m < 2$ are impossible, because for $m = 0$ our equality turns into a false equality $2 = 3$, and for $m = 1$ it turns into an equality $1 - z + 1 + z = 2 + z$, which is true only for $z = 0$. So $m \geq 2$.

Since both sides are analytical in z, the second derivatives of both sides at $z = 0$ must be equal to each other. The second derivative of the left-hand side at $z = 0$ is equal to $m(m-1)$. The second derivative of the right-hand side is equal to $2m(m-1)z^{m-2}$. If $m > 2$, then this derivative equals 0 at $z = 0$ and therefore cannot be equal to $m(m-1)$. So $m \geq 2$, and m cannot be greater than 2. Therefore, $m = 2$.

So, $l(x) = Bx^2$, and hence $\mu(x) = exp(-\beta x^2)$ for some $\beta > 0$. Q.E.D.

6. CONCLUSIONS

How to represent in mathematical terms uncertain numeric statements about the value x of a physical quantity, e.g., statements of the type "most likely x is between $a - \delta$ and $a + \delta$"? Reasonable arguments lead us to the conclusion that the most adequate membership functions for such statements are Gaussian functions $\mu(x) = exp(-\beta(x-a)^2/\delta^2)$.

If we use these membership functions, then we can apply simple algorithms to combines the opinions of several experts. Namely, if k experts give estimates $a_1, ..., a_k$, and they estimate the precision of their estimates as correspondingly $\delta_1, \delta_2, ..., \delta_k$, then the resulting membership function is equal to $\mu(x) = exp(-\beta(x-a)^2/\delta^2)$, where $\delta = (\delta_1^{-2} + \delta_2^{-2} + ... + \delta_k^{-k})^{-1/2}$ and

$$a = (a_1\delta_1^{-2} + ... + a_k\delta_k^{-2})/(\delta_1^{-2} + ... + \delta_k^{-2}).$$

These formulas coincide with the ones that result from applying the statistical least squares method, so we do not even have to write a new software.

This approach was applied to testing jet engine for aircraft and spaceships, and it may be useful in many other applications, e.g., in combining the results of several coordinate and distance sensors in spaceship navigation.

ACKNOWLEDGEMENTS

This work was started with the support of the Soviet Space Shuttle Program and was continued under NSF Grant No. CDA-9015006, NASA Research Grant No. 9-482, and the Institute for Manufacturing and Materials Management grant. The authors are greatly thankful to Harold Brown (GE Aircraft Engines) and Bob Lea (NASA Johnson Space Center) for inspiring discussions.

REFERENCES

[A66] J. Aczel. *Lectures on functional equations and their applications.* Academic Press, N.Y. and London, 1966.

[BCDMMM85] G. Bartolini, G. Casalino, F. Davoli, M. Mastretta, R. Minciardi, and E. Morten. *Development of perfomance adaptive fuzzy controllers with applications to continuous casting plants*, in: M. Sugeno (editor). *Industrial applications of fuzzy control*, North Holland, Amsterdam, 1985, pp. 73–86.

[HC76] H. M. Hersch and A. Caramazza. *A fuzzy-set approach to modifiers and vagueness in natural languages.* J. Exp. Psychol.: General, 1976, Vol. 105, pp. 254–276.

[K75] A. Kauffman. *Introduction to the theory of fuzzy subsets. Vol. 1. Fundamental theoretical elements*, Academic Press, N.Y., 1975.

[KQLFLKBR92] V. Kreinovich, C. Quintana, R. Lea, O. Fuentes, A. Lokshin, S. Kumar, I. Boricheva, and L. Reznik. *What non- linearity to choose? Mathematical foundations of fuzzy control.* Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference, Louisville, KY, 1992, pp. 349–412.

[KM87] R. Kruse and K. D. Meyer. *Statistics with vague data.* D. Reidel, Dordrecht, 1987.

[O77] G. C. Oden. *Integration of fuzzy logical information*, Journal of Experimental Psychology: Human Perception Perform., 1977, Vol. 3, No. 4, pp. 565–575.

[KR86] V. Kreinovich and L. K. Reznik. *Methods and models of formalizing prior information (on the example of processing measurements results).* In: *Analysis and formalization of computer experiments*, Proceedings of Mendeleev Metrology Institute, Leningrad, 1986, pp. 37-41 (in Russian).

[YIS85] O. Yagishita, O. Itoh, and M. Sugeno. *Application of fuzzy reasoning to the water purification process*, in: M. Sugeno (editor). *Industrial applications of fuzzy control*, North Holland, Amsterdam, 1985, pp. 19–40.

[Z65] L. Zadeh. *Fuzzy sets.* Information and control, 1965, Vol. 8, pp. 338–353.

[Z78] H. J. Zimmermann. *Results of empirical studies in fuzzy set theory.* In: *Applied General System Research* (G. J. Klir, ed.) Plenum, New York, 1978, pp. 303–312.

Paper Not Submitted

in Time for Publication

N93

29585

UNCLAS

Life Insurance Risk Assessment Using a Fuzzy Logic Expert System

Luis A. Carreno and Roy A. Steel
Togai Infralogic, Inc.

Abstract

In this paper, we present a knowledge based system that combines fuzzy processing with rule-based processing to form an improved decision aid for evaluating risk for life insurance.

This application illustrates the use of FuzzyCLIPS to build a knowledge based decision support system possessing fuzzy components to improve user interactions and KBS performance. The results employing FuzzyCLIPS are compared with the results obtained from the solution of the problem using traditional numerical equations. The design of the fuzzy solution consists of a CLIPS rule-based system for some factors combined with fuzzy logic rules for others. This paper describes the problem, proposes a solution, presents the results, and provides a sample output of the software product.

1.0 Introduction to FuzzyCLIPS

FuzzyCLIPS adds fuzzy processing capability to CLIPS 5.1. The architecture is a separate processing element similar to that used to incorporate object-oriented programming into CLIPS. The basic fuzzy constructs and function calls can be written intermixed with usual CLIPS statements. Principal fuzzy constructs define rule bases and membership functions. A fuzzy membership function can be associated with a universe of discourse. This improvement allows readable terms such as "high" and "low" t be used in different contexts. There are also functions by which a CLIPS program can test the degree of membership of a sensor value, execute a fuzzy rule base that returns defuzzified control values to CLIPS and, optionally, assert facts giving belief values for the possibilities that might be useful in an expert system. In addition, C interface functions support embedded fuzzy applications that can invoke the fuzzy processor directly for speed in embedded control applications. FuzzyCLIPS is designed to be compatible with future CLIPS versions. Like CLIPS, it can operate as a stand alone program or be embedded in a larger application.

2.0 Problem Statement

An insurance company needs to assess the degree of health risk associated with each client based on physical characteristics such as height, weight, and age and exercise, smoking, drinking, and eating habits. The output risk value serves as the basis for the determination of insurance premiums billed to clients. Those premiums have a base rate (perfect health, good habits, 35 years old) and an increment to adjust the premium based on the risk. A system that produces a risk value between 0.0 and 1 suffices to set a net rate. The equation is

Cost to Insure
Client = Base Rate + ((Risk /Base Risk)-1)*Increment (1)

The relation between decision factors and the rate change need be neither incremental nor linear, i.e., separate consideration of the decision factors may not determine a change in rate that can be simply summed to determine the net rate. This means that the questioning of the client must be controlled; it makes no sense to continue to ask a client about all factors if a decision on rates is possible at some intermediate point in the interaction. Complex nonlinearity and interdependence of the factors mean that computer-based decision aids are useful to a human agent and that sharp decision boundaries such as those produced by a normal rule based system are sensitive to small uncertainties in the input data. Fuzzy logic provides a basis for accommodating such uncertainty with finesse.

The input variables of the system are of two different types: base and incremental. The base type of input variables are Age (A), Weight (W), and Height (H). A derived internal variable is the body mass index (BMI) that estimates fitness or body fat content. Incremental input variables deal with particular habits and characteristics of prospective clients. The following are considered such variables in the present example exercising (E), dairy products intake (DI), red meat intake (MI), vegetable intake (VI), fat/sweet intake (FSI), smoking (S), and drinking (D). The output of the system is the degree of risk (R).

3.0 Traditional Numerical Solution

For the traditional method solution, we treat all of the variables as a number input or a selection from a finite, discrete, closed set of possibilities. Each variable is represented as a lookup table of intervals where the value of the corresponding is specified for each interval. The following table presents the values of the contribution to risk due to Age,

age	age-risk
0 to 30	0.25
31 to 60	0.5
61 to 90	0.75
>90	1.0

We note that this table could be used in a rule-based knowledge system (KBS) to provide rules of the form

```
(age ?age&:( <= ?age 30) => (assert (age-risk .25))

(age ?age1&:( > ?age1 30))
(age ?age2&:(<= ?age2 60))(test (!= ?age1 ?age2))
 => (assert (age-risk = 0.5))
etc.
```

For discrete selections, the table contains the risk value assigned to each value. An example corresponding rule is

```
(smoke-habit ?input&((eq ?input O)||(eq ?input S)...))
 => (assert (smoke-factor 0.25))
```

When each factor has been evaluated, the total risk is evaluated as a weighted combination of the risks due to various factors where the value of the weights provide another knowledge component of the decision support system.

3.1 Body Mass Index

The inputs, Height and Weight, are used to obtain the body mass index (BMI). This measure determines if a person is overweight or not. BMI is calculated by dividing the Weight in kilograms by the square of the height in meters, BMI = Weight/(Height)^2.

The following table shows the scale used to measure BMI and the corresponding BMI-risk that is used later to calculate risk.

BMI	Condition	BMI-risk
under 23	Underweight	0.25
23 -25	Ideal	0.0
25 - 30	Overweight	0.75
over 30	Obese	1.0

3.2 Mathematical Model for Traditional System

In a traditional system, the first step in the solution of the problem is to define a mathematical relation between the inputs and outputs of the system. The objective is to obtain a numerical value that represents the risk of a person having medical problems due to his physical characteristics and eating habits. Risk is defined as having a range of [-0.357,1]. The various factors are also assumed to have values in the [0,1] range by mappings similar to those presented above for age and BMI. A risk measure of 1 represents the maximum degree of risk, on the contrary, a measure of 0 or less represents the minimum degree of risk.

In general,

$$\begin{aligned}\text{Risk} = {} & w_{BMI}*(\text{BMI-risk}) + w_S*(\text{Smoking-risk}) + w_D*(\text{Drinking-risk}) + \\ & w_E*(\text{Exercise-risk}) + w_{VI}*(\text{Vegetarian-risk}) + w_{DI}*(\text{Dairy-Products-}\\ & \text{Intake-risk}) + w_{MI}*(\text{Red-Meat-Intake-risk}) + w_{FSI}*(\text{Fat/Sweet-Intake-}\\ & \text{risk}) + w_A*(\text{Age-risk}) \qquad (2)\end{aligned}$$

Constants w_E and w_{VI} are negative because they reduce total risk. The other weights are expected to be positive. Values of the weights are based on the corresponding factor's effect on the overall degree of health of a person.

3.3 Effects of Habits (Incremental Inputs)

In addition to age and BMI, factors reflecting a person's habits contribute to risk assessment. These are generally harder to quantify and are often described by qualitative terms such as "I smoke a little" or "I eat lots of vegetables." There are two approaches that are used to handle such data. The normal one is to attempt to quantify the habit in terms of frequency of participation and quantity of material, time, or activity concerned, much as the scientist who studies effects of various habits on health risk quantifies inputs to the evaluation experiments. The other approach is to classify estimates of activity frequency and level into literal categories from options to the respondent. For example, exercise might be analyzed from a more complicated user interface

Level
_aerobic
_strength building
_other
_active work
_do not exercise

Frequency
_very frequent
_frequent
_ sometimes
_occasionally
_never

Type
_walking or treadmill
_jogging
_lift weights
_exercise machine
_water sports
_team sports
_skating
_skiing

The disadvantage of such an approach is that a need for understanding the respondent's meaning for a term means ambiguity in the input data and stress for the respondent in deciding which category fits his case. In general, more complex interfaces are required to provide sufficient detail or correlations from which to extract information about whether the user understands or is trying to bias answers in his favor.

A user interface in which the user chooses values for frequency and intensity against an arbitrary scale (e.g., "on a scale of 1 to 10, how much do you drink?) introduces the potential to fuzzify the input to conduct reasoning with correlation and interpolation between benchmarks or way points.

Qualitative values indicating the change in risk due to various habits is shown below.

	Health Risk		
	Risk Increases	Neutral	Risk Decreases
Smoking	High	Med	None
Drinking	High	Med	None
Exercising	Low	Med	High
Vegetable Intake	Low	Med	High
Red Meat Intake	High	Med	Low
Dairy Intake	High	Med	Low
Fat/Sweet Intake	High	Med	Low

4.0 Fuzzy Logic Solution

In a fuzzy logic based system, an expert defines the rules. Such rules are used to describe the characteristics of the risk assessment for each factor. Later on, the input variables are matched against the set of rules to produce the appropriate output. Each one of the fuzzy variables contributes to the output of the system depending on how many rules are fired for each particular input variable. Fig. 1 depicts a schematic fuzzy decision support system. For fuzzy reasoning we use a max-dot inferencing technique, and centroid defuzzification technique.

For this particular example, four different sets of fuzzy rules are defined. The first rulebase relates a risk_1 to age and BMI. The second rulebase relates a risk_2 to smoking and drinking habits. The third rulebase relates a risk_3 to the amount of exercise and intake of vegetables. The last rulebase relates a risk_4 to intake of dairy products, red meat, and fat and sweet products. A fifth rulebase relates risks 1-4 to the overall risk to complete the risk assessment. The importance of breaking down the problem into smaller related groups is the fact that the number of rules needed to control the system decreases dramatically. In our example, the number went down from 4* 3^7 (8748) rules to a maximum of 313 rules.

After calculating the BMI and having obtained the age from the user interface, an initial measure of risk, risk_1, is obtained. This measure serves as the basis for subsequent decisions. If the risk obtained is considered by the system as very high, no further inquiries of the user are necessary. On the other hand, if the risk obtained is considered low, medium, or high, further inquiries into the client's habits are necessary to produce a more meaningful result.

The output of the system consists of a crisp value for Risk in the range [0, 1]. The system also produces a truth value associated with each output fuzzy set, i.e., the degree to which each fuzzy set defining risk contributes to the output value of risk.

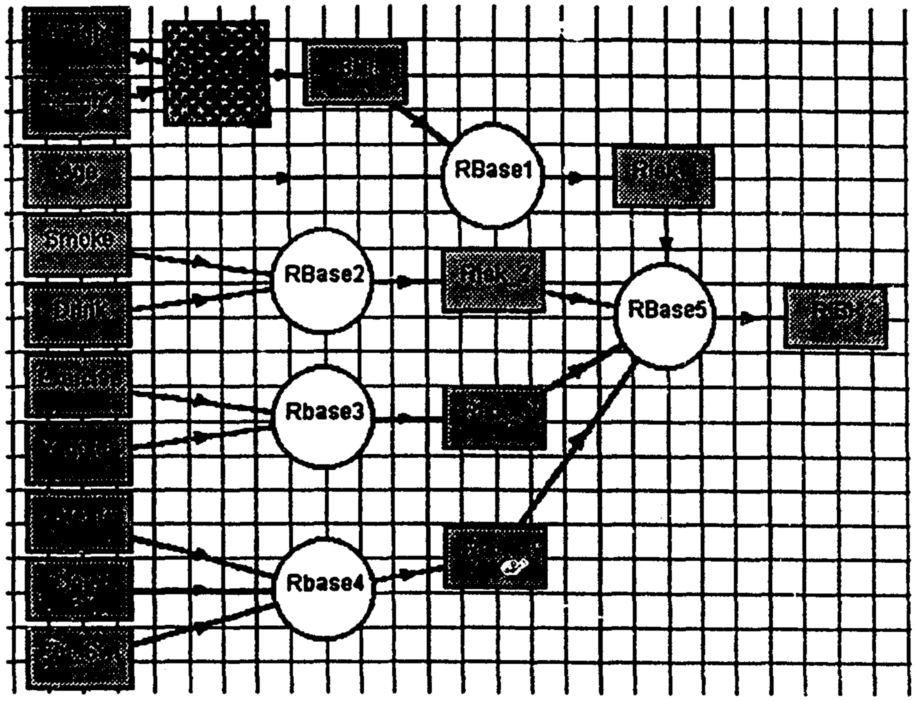


Fig.1 A schematic view of the fuzzy logic risk assessor

4.1 Membership Functions

In order to solve the problem using fuzzy logic methods, we defined sets of membership functions associated with each variable

A	Lo, Med, Hi
BMI	Under, Ideal, Over, Obese
Risk_n	Low, Medium, High, Very High

The universe of discourse for each of the above fuzzy variables is [0,1] for each risk (Fig. 2), [0,40] for BMI (Fig. 3), and [0,100] for Age (Fig. 4).

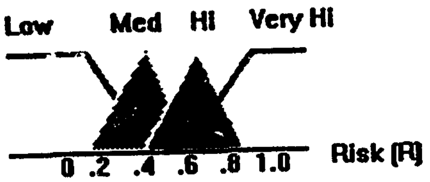


Fig. 2 Risk Membership Functions

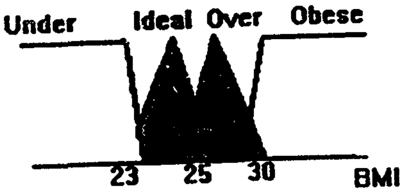


Fig. 3 BMI Membership Functions

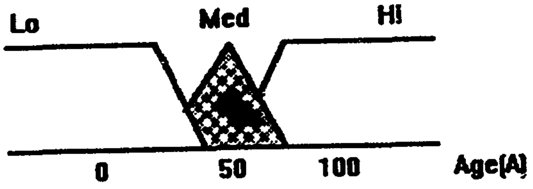


Fig. 4 Age Membership Functions

4.2 Rules

A sample of the fuzzy logic rule set for Risk, based on all the inputs as a whole, can be seen in the following table.

RISK	A	BMI	E	VI	DI	MI	FSI	S	D
Low	Lo	Ideal	Med	Hi	Med	Lo	Lo	Lo	Lo
	Lo	Ideal	Hi	Med	Lo	Lo	Lo	Lo	Lo
	Lo	Ideal	Lo	Hi	Lo	Lo	Med	Lo	Lo
Medium	Med	ideal	Lo	Lo	Hi	Hi	HI	Lo	Lo
	Med	Over	Med	Med	Lo	Lo	Med	Med	Lo
	Lo	Over	Lo	Lo	Med	Hi	Hi	Lo	Med
High	Med	Obese	Lo	Hi	Lo	Lo	Hi	Hi	Lo
	Med	Over	Lo	Lo	Med	Hi	Hi	Hi	Med
	Hi	Over	Lo	Lo	Hi	Med	Med	Med	Med
Very High	Med	Obese	Lo	Lo	Lo	Hi	Hi	Med	Hi
	Hi	Obese	Lo	Lo	Hi	Hi	Hi	Hi	Med
	Hi	Over	Lo	Lo	Hi	Hi	Hi	Hi	Hi

As explained earlier, a rulebase with that many inputs is difficult to implement due to the large number of possible combinations of the input variables. Examples of fuzzy rules, using the alternative approach of breaking down the input variables into smaller and related groups, is shown next.

IF A is Hi and
BMI is Obese
THEN **Risk_1** is Very High

IF S is HI and
D is L
THEN **Risk_2** is High

IF E is Hi and
VI is M
THEN **Risk_3** is Low

IF MI is L and
DI is M and
FSI is M
THEN **Risk_4** is Medium

In the application, five rulebases are defined. As explained earlier, each one produces a partial risk that is merged at the end of processing to produce a final assessment of risk. Such risk is compared with an ideal risk called base risk. The base risk is the risk associated with a 35 year old with the following physical characteristics and drinking/eating habits ideal BMI, non smoker, low consumption of alcoholic drinks, low consumption of dairy products, red meat products, and fat/sweet products, high consumption of vegetables, and high amounts of exercise. The total risk of a particular person is calculated and substituted in Eq. (1) to produce a premium amount.

4.3 User Interface

There are two special cases in the processing of the problem. First, if the initial risk, based on age and BMI, is greater than 0.8 the risk is considered very high. Therefore, no need for further processing of the system. Second, if the initial assessment of BMI is greater than 30, meaning the person is obese, questions related to the habits of consumption of dairy products, red meat, and fat/sweet products are omitted. Otherwise, the user interface is the same as that for the numerical method.

5.0 Results and Conclusions

To compare the methods, sample data was created and processed by both versions of the program. The sample data consists of a group of persons with the same eating and exercise habits, the only variant is the age of the individuals. The constant characteristics can be seen in the following table.

BMI	S	D	VI	ESI	E	DI	MI
Ideal	no	L	H	L	H	L	L

The values of age used were in the range [20, 100]. The results were as expected. For the traditional method, we can see abrupt changes in the value of risk associated with ages at the edges of the intervals, as observed in figure 5, the value of risk jumps from age 30 and then continues constant until it reaches the age of 60 where it jumps again. The process is repeated at age 90.

For the fuzzy logic solution, as observed in figure 5, no sharp differences are produced at any specific age, i.e., the values of risk increase smoothly along the whole universe of discourse. The fuzzy system produces more realistic values for different ages, specially for those cases in which the age varies from 30 to 31, 60 to 61, or 90 to 91.

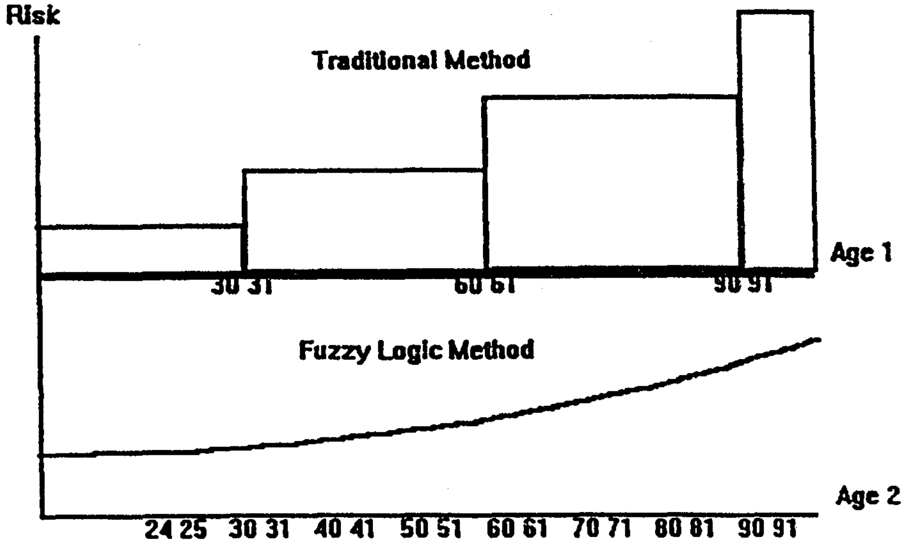


Fig. 5 Risk for Traditional Vs Fuzzy Logic Method

6.0 Sample Output

The application program described in this document, was written using the alpha version oi FuzzyCLIPS. It generates an interactive session, in which the user is questioned in order to gather information about a client's physical characteristics, exercise habits, and eating and drinking habits.

After receiving all of the information needed, the partial values of risk are determined, and a final summary report is produced. It consists of the four partial risks and its values, the total value of risk, the value of the base risk, explained earlier, the ratio of the total to base risk, the annual insurance premium, and the individual contributions of each membership function by risk and its predicate values.

```
*************************************************
                  SUMMARY
*************************************************
Risk based on
------------------------------------
age and bmi ================>  0.318
smoking/drinking ============> 0.600
exercise/vegetable intake =======> 0.400
fat intake ===================>  0.842
*************************************************
                     total risk =>  0.547
          BASE RISK => 0.344
*************************************************
RATIO total/base risk => 1.59
*************************************************
  YOUR ANNUAL PREMIUM IS ===>  $ 1941.13
*************************************************
     INDIVIDUAL MBF CONTRIBUTIONS BY RISK
*************************************************
Fat intake Risk ===> MBF VH      Degree of Truth 1.0
Exer/Veggies Risk =>MBF H        Degree of Truth 4.2e-005
Exer/Veggies Risk =>MBF M        Degree of Truth 0.99
Smoke/Drink Risk ==>MBF H        Degree of Truth 0.99
Smoke/Drink Risk ==>MBF M        Degree of Truth 4.2e-005
Age/BMI Risk ====> MBF M         Degree of Truth 0.587
Age/BMI Risk ====> MBF L         Degree of Truth 0.412
**************************************************
```

7.0 References

1. FuzzyCLIPS Reference Manual, Vol. 1, Basic Programming Guide, Alpha Release, Oct. 19, 1992.

2. CLIPS Reference Manual, Vol 1, Basic Programming Guide JSC-25012, NASA/JSC Sept. 10, 1991.

3. The New Good Housekeeping Family Health and Medical Guide, section three, p. 608-615, (Hearst Corporation, New York, 1989).

Omit to End

Aazhang, Behnaam 589
Barone, Joseph M. 205
Bezdek, James C. 98
Bogdan, I. 350
Bosc, Patrick 478
Buhusi, Catalin V. 360
Carreno, L. A. 627
Chang, Ching-Chuang 418
Chen-Kuo Tsao, Eric 98
Chiang, Jung-Hsien 257
Chiu, Stephen 239
Chung-Hoon Rhee, Frank 81
Cohen, M. E. 535
Copeland, Charles 154
Cutello, Vincenzo 215
Dasarathy, Belur V. 368
Daugherity, Walter C. 143
Dekorvin, Andre 398
Deutsch-McLeish, Mary 266
Dockery, John T. 248
Filev, Dimitar P. 135
Fleischman, Robert M. 496
Fleming, J. W. 524
Flikop, Ziny 378
Frigui, Hichem 59
Gader, Paul 257
Gantner T. E. 427
Gebhardt, Jorg 296
Gorzalczany, Marian B. 266
Grantner, Janos 312
Gutierrez-Martinez, Salvador 437
Henson, Troy F. 589
Holden, A. V. 545
Hu, Yong Lin 488
Huang, Song 108
Huang, X. H. 524
Hudson, D. L. 535
Huxhold, Wendy L. 589
Janabi, Talib H. 340, 449
Joslyn, Cliff 458
Juang, C. H. 524
Karpovsky, Efin Ja. 224
Karr, C. Lucas 506
Karwowski, Waldemar 1
Keller, James M. 49
Keller, Jim 468
Klir, George 598
Koczy, Laszlo T. 608
Kraslawski, A. 408
Kreinovich, V. 418, 618
Krishnapuram, Raghu 49, 59, 81
Krowidy, Srinivas 488
Kruse, Rudolf 296, 388
Labos, A. S. 545
Labos, E. 545
Laczko, J. 545
Lara-Rosano, Felipe 304
Lea, Robert N, 154, 398, 515
Lee, Hon-Mun 125
Lee, Jonathan 185
Leigh, Albert B. 69
Lim, P. Y. W. 427
Littman, David 248
Ma, Yibing 49
Madyastha, Raghavendra K. 589
Mansfield, W. H. 496
McAllister, Luisa 555
Mitra, Sunanda 166
Mohamed, Magdi 257
Montero, Javier 215
Nair, Satish S. 118
Nasraoui, Olfa 59
Natarajan, Swami 185
Nauck, Detlef 296, 388
Niskanen, Vesa A. 625
Nolan, Adam 488
Nystrom, L. 408
Ostaszewski, Krzysztof 1
Pagni, Andrea 195
Pal, Nikhil R. 98
Pal, Sankar K. 69
Parlos, Alexander 125
Parviz, Behzad 598
Patyra, Marek J. 312, 565
Pavel, Sandy 575
Pedrycz, W. 322, 581
Pemmaraju, Surya 166
Pfluger, Nathan 185
Pivert, Olivier 478
Poluzzi, R. 195
Prade, Henri 7
Quintana, C. 618
Ramamoorthy, P. A. 108
Ramer, Arthur 14
Reznik, L. 418, 618
Rizzotto, G. G. 195
Roventa, Eugene 581
Rueda, A. 322
Ryjov, Alexander P. 21
Schott, Brian 330
Seniw, David 175
Shehadeh, Hana 515
Shieh, C-Y. 118
Shipley, Margaret F. 398
Solopchenko, G. N. 418
Stachowicz, Marian S. 312
Steel, R. A. 627
Steinlage, Ralph C. 427

Sultan, L. H. 340
449

Sztandera, Leszek M. 89

Tahani, Hossein 468

Tavakoli, Nassrin 175

Teodorescu, H. N. 350

Tian, Y. 29

Tsai, Wei K. 125

Turksen, I. B. 29
276
286

Vaidya, Nitin 39

Villarreal, James A. 154

Wang, Haojin 143

Wee, Willaim G. 488

Whalen, Thomas 229
330

Willson, Ian A. 276
286

Yager, Ronald R. 7

Yashvant, Jani 154

Yen, John 39
143
185

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1992

3. REPORT TYPE AND DATES COVERED
Conference Publication

4. TITLE AND SUBTITLE
NAFIPS '92 North American Fuzzy Information Processing
Society Volume 2

5. FUNDING NUMBERS

6. AUTHOR(S)
James Villarreal, compiler (NASA)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lyndon B. Johnson Space Center
Houston, Texas 77058

8. PERFORMING ORGANIZATION REPORT NUMBER
S-702

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
CP 10112

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Unlimited/Unclassified

Subject Category 59

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document contains papers presented at the NAFIPS '92 North American Fuzzy Information Processing Society Conference, held at the Melia Hotel Paseo de la Marina Sur Marina Vallarta in Puerto Vallarta, Mexico, on December 15-17, 1992. More than 75 papers were presented at this Conference, which was sponsored by NAFIPS in cooperation with NASA, the Instituto Tecnologico de Morelia, the Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP), the Instituto Tecnologico de Estudios Superiores de Monterrey (ITESM), the International Fuzzy Systems Association (IFSA), the Japan Society for Fuzzy Theory and Systems, and the Microelectronics and Computer Technology Corporation (MCC).

The fuzzy set theory has led to a large number of diverse applications. Recently, interesting applications have been developed which involve the integration of fuzzy systems with adaptive processes such as neural networks and genetic algorithms. NAFIPS '92 was directed toward the advancement, commercialization, and engineering development of these technologies.

14. SUBJECT TERMS
Fuzzy Systems, Neural Networks, Genetic Algorithms, Optimization, Pattern Recognition, Path Planning, Robotics, Information Processing and Vision, Decision Analysis, Control Systems

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
Unlimited

DO NOT REMOVE SLIP FROM MATERIAL

Delete your name from this slip when returning material to the library.

NAME	DATE	MS
M. Nemeth	1-98	280

NASA Langley (Rev. Dec. 1991) RIAD N-75