Collected Notes on the Workshop for Pattern Discovery in Large Databases (NASA Ames, January 14-15, 1991)

Wray Buntine and Martha Del Alto
RIACS and Recom Technologies
AI Research Branch, Mail Stop 244-17
NASA Ames Research Center
Moffett Field, CA 94035

NASA Ames Research Center
Artificial Intelligence Research Branch

Technical Report FIA-91-07

April, 1991

Collected Notes on the Workshop for Pattern Discovery in Large Databases[1] (NASA Ames, January 14-15, 1991)

edited by

Wray Buntine and Martha Del Alto
RIACS and Recom Technologies
NASA Ames Research Center
Mail Stop: 244-17
Moffett Field, CA 94035

April, 1991

---

1 The organization of this workshop and the preparation of this report have been jointly supported by NASA code FI (Information Sciences Division) and the Research Institute for Advanced Computer Science (RIACS).

Contents

Introduction and Overview

Many scientists are noticing the large amount of data that is becoming available for for analysis and the need for methods to extract useful information out of the data. The fields of "knowledge discovery" in Artificial Intelligence and related areas, "database mining" in information systems and neural networks, and of course statistics and pattern recognition are concerned with these issues.

NASA missions produce large volumes of data that need analysis, for instance, proposals such as the Earth Observing System, and various deep space missions. Data analysis researchers at Information Sciences Division of NASA Ames Research Center and RIACS organized the Pattern Discovery in Large Databases Workshop to bring into focus the data analysis research requirements of government institutions like NASA, and to gather together researchers from a wide spectrum of data analysis methodologies and applications: statistics, neural networks, artificial intelligence, databases, medical, biomedical, remote sensing and astrophysical applications, etc.

The workshop was held at NASA Ames Research Center on 14-15 January. It addressed the core data analysis tasks that have traditionally required statistical or pattern recognition techniques. Some of the core tasks include classification, discrimination, clustering, supervised and unsupervised learning, discovery and diagnosis, i.e. general pattern discovery.

There were two panels at the workshop. The first discussed advanced research topics such as model-based analysis and analysis of data from multiple sources. The second panel was a "wrap-up" where views were aired concerning the requirements of research, issues to be resolved, the state of the art, etc. Issues discussed ranged from software development, data visualization, interaction with the domain expert, methodologies for analysis, to interpretation, calibration, and control of heterogeneous and multiple source data, etc.

These collected notes are a record of material presented at the workshop. A second report is now being prepared to give the organizers' perception of the research requirements and issues involved in pattern discovery of large databases, as expressed by participants at the workshop.

Workshop on Pattern Discovery in Large Data Bases

Sponsored by NASA, Information Sciences Division, and RIACS[1]

January 14-15, 1991
NASA Ames Research Center
Moffett Field, California

NASA missions produce large volumes of data that need analysis. Recent proposals, such as the Earth Observing System, MTPE and the Exploration Technology Initiative bring into focus the need for more powerful techniques for analysing large data sets. This workshop is the first in a series planned by the Information Sciences Division (code FI) at NASA Ames intended to focus attention on underlying problems in this analysis and to establish working relations between the various NASA and other government centers concerned with these problems.

This workshop will address the core data analysis tasks that have traditionally required statistical or pattern recognition techniques. Recent developments in statistics, artificial intelligence, and neural networks offer methods for this analysis. Some of the core tasks include classification, discrimination, clustering, supervised and unsupervised learning, discovery and diagnosis, i.e. general pattern discovery. For instance, unsupervised learning in a remote sensing application typically involves trying to identify groups of related spectral patterns that might possibly correspond to distinct vegetation types or land use. Supervised learning involves the additional information of ground truth that is used in conjunction with the spectral data to develop ways of predicting land use on future data for which ground truth is not available. These two tasks broadly correspond to making sense out of the existing data, and making predictions on future data.

This workshop is intended to bring together active researchers in the data analysis area: representatives from agencies performing data analysis including NASA, NIH, NOAA, NCAR, etc. Representatives from the major methodologies such as AI, statistics and neural networks will also be represented. The aim of the workshop is to initiate a coordinated effort on data analysis. We hope to clarify the requirements of large database analysis and coordinate research in data analysis methodologies.

Workshop Format

The workshop is by invitation and is expected to involve 20-30 participants. Representatives are being invited from agencies such as NIH, NCAR (National Center for Atmospheric Research), from the astronomy and medical communities and NASA. Participants will be given opportunity to present their work along with time for group discussion.

Arrangements

The workshop will be held at the NASA Ames Research Center, California. Attendees will be expected to make their own arrangements, although information will be provided regarding local accommodation and services. Special arrangements will have to be made at least one month in advance for non-residents of the US. Local details will be provided on indication of intent to attend the workshop.

Contacts

Those interested in attending the workshop should contact one of the organisers below. Those interested in giving a talk should send a small abstract (1/2 page) with references to related work.

Wray Buntine	Peter Cheeseman	Chuck Jorgensen
		Center for Advanced Data Evaluation Tech.
NASA Ames Research Center	NASA Ames Research Center	NASA Ames Research Center
MS 244-17	MS 244-17	MS 244-4
Moffett Field, CA, 94035	Moffett Field, CA, 94035	Moffett Field, CA, 94035
(415) 604 3389	(415) 604 4946	(415) 604 6725, (FTS) 464 6725
wray@ptolemy.arc.nasa.gov	cheeseman@pluto.arc.nasa.gov	jorgensen@pluto.arc.nasa.gov

[1] Research Institute for Advanced Computer Science

Program for the Workshop on Pattern Discovery in Large Data Bases

Organized by RIACS and Information Sciences Division at NASA Ames Research Center

January 14-15, 1991
Auditorium, Bldg 245, NASA Ames Research Center

Day 1 :

Start	8:30	*Coffee and snacks*
	9:00- 9:15	**Welcome/Opening Remarks**
	9:15 - 9:50	**On the Extraction of Information from Multispectral Image Data** David Landgrebe, Purdue University
	9:50 - 10:25	**Bayesian Inference Applied to Large Data Bases** Peter Cheeseman, RIACS and NASA FIA
Break	10:25 - 10:45	*Coffee and snacks*
	10:45 - 11:20	**Knowledge Acquisition Planning** Lawrence Hunter, National Library of Medicine
	11:20 - 11:55	**Aspects of Astronomical Research Involving Large Data Bases** Nick Weir and Stan Djorgovski, California Institute of Technology
	11:55 - 12:30	**Interactive or Automatic?** Creon Levitt, Al Globus and Steve Bryson, NASA NAS and Sterling Federal Systems
Lunch	12:30 - 2:00	*On your own*
	2:00 - 2:35	**Smooth Mixture Estimation with Multichannel Image Data** John McDonald and Finbarr O'Sullivan, University of Washington
	2:35 - 3:10	**Automated Causal Inference from Large Data Bases** Peter Spirtes, Richard Scheines and Clark Glymour, Carnegie Mellon University
	3:10 - 3:45	**A Neural Network to Extract Implicit Knowledge from a Nuclear Database,** Awatef Gacem, Aliana Maren, and Robert Uhrig, University of Tenessee
	3:45 - 5:30	**Poster Session - Wrap Up** (*with coffee*)

Day 2 :

Start	8:30	*Coffee and snacks*
	8:55 -9:00	**Day 2 Welcome**
	9:00- 10:10	**Model-Based Data-Analysis** Panel
	10:10 - 10:45	**A Bayesian Method for the Induction of Probabilistic Networks from Data** Gregory F. Cooper, University of Pittsburgh and Edward Herskovits, Stanford University
Break	10:45 - 11:05	*Coffee and snacks*
	11:05 - 11:40	**A Strategy For Large-Area Land Cover Characterization** Thomas R. Loveland, US Geological Survey, James Merchant, Uni. of Nebraska-Lincoln and Donald Ohlen, TGS Tech. Inc.
	11:40 - 12:15	**Remote Sensing for Ecosystem Monitoring** Chris Hlavka, NASA SGE
	12:15 - 12:50	**Applications of Scale-Space Filtering and Labyrinth to Soil Analysis** Deepak Kulkarni and Kevin Thompson, NASA FIA and Sterling Federal Systems
Lunch	12:50 - 2:00	*On your own*
	2:00 - 3:00	**Analysis Using Multiple Data Bases** Panel
	3:00	Workshop Wrap Up

Workshop Participants

Rakesh Agrawal
IBM Almaden Research Center
K54/802
650 Harry Rd.
San Jose CA 95120

Wray Buntine
RIACS and
Artificial Intelligence Research Branch
MS 244-17
NASA Ames Research Center

Penny Chase
The MITRE Corporation
Mail Stop A040
Burlington Road
Bedford, MA 01730

Peter Cheeseman
RIACS and
Artificial Intelligence Research Branch
MS 244-17
NASA Ames Research Center

Stuart Crawford
Advanced Decision Systems
1500 Plymouth Street
Mountain View, CA, 94043-1230

Peter Denning
RIACS
625 Ellis Street, Suite 205
Mountain View, CA, 94043

Susan Eberlein
Jet Propulsion Laboratory, 168-522
Pasadena, CA 91109, USA

Jeff Eidenshink
TGS Technology, Inc.
Sioux Falls, SD, 57198

Usama Fayyad
Uni. of Michigan
P.O. Box 4308
Ann Arbor, MI 48106

Bob Fung
Advanced Decision Systems
1500 Plymouth Street
Mountain View, CA, 94043-1230

Len Gaydos
SGE & USGS
MS 242-4
NASA Ames Research Center

Sheri Gish
IBM Almaden Research Center
K54/802
650 Harry Rd.
San Jose CA 95120

Hayit Greenspan
Jet Propulsion Laboratory, 168-522
Pasadena, CA 91109, USA

Edward Herskovits
Knowledge Systems Lab
Medical school Office Building X-215
Stanford, CA, 94305-5479

Chris Hlavka
SGE
MS 242-4
NASA Ames Research Center

Lawrence Hunter
National Library of Medicine
Lister Hill Center, MS-54
Bethesda, MD 20894

Louis Jaeckel
RIACS
Mail Stop: Ellis
NASA Ames Research Center

Charles Jorgensen
Intelligent Systems Technology Branch
MS 244-4
NASA Ames Research Center

Deepak Kulkarni
Artificial Intelligence Research Branch
MS 244-17
NASA Ames Research Center

David Landgrebe
School of Elect. Eng.
Purdue University
West Lafayette, 47907

Creon Levitt
NAS
MS T045-1
NASA Ames Research Center

Danika Lew
Dept. of Statistics
University of Washington
Seattle, WA 98195

Steve Litvintchouk
The MITRE Corporation
Mail Stop A150
Burlington Road
Bedford, MA 01730

John MacDonald
Dept. of Statistics
University of Washington
Seattle, WA 98195

Alianna Marek
University of Tennessee
Knoxville, TN 37996

Charles Miles
Recom Technologies
8321 Auburn Avenue, Suite 165
Citrus Heights, CA 95610

Finbarr O'Sullivan
Dept. of Statistics
University of Washington
Seattle, WA 98195

Dragutin Petkovic
IBM Almaden Research Center
K54/802
650 Harry Rd.
San Jose CA 95120

Gregory Piatetsky-Shapiro
GTE Laboratories Incorporated
40 Sylvan Road, MS-45
Waltham, MA 02254

Padhraic Smyth
Communication Systems Research
Jet Propulsion Laboratory, 238-420
Pasadena, CA 91109, USA

Peter Spirtes
Lab. for Computational Linguistics
139 Baker Hall
Dept. of Philosophy
Carnegie Mellon University
Pittsburgh, PA, 15213

David States, M.D.-Ph.D.
Nat. Ctr. for Biotechnology Information
Naitonal Library of Medicine
Building 38A, Room 8S806
8600 Rockville Pike
Bethesda, MD 20894

Kevin Thompson
Artificial Intelligence Research Branch
MS 244-17
NASA Ames Research Center

Nick Weir
California Institute of Technology

Real Time Multispectral Pattern Recognition

Susan Eberlein, Gigi Yates, and Eric Majani

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, MS 168-522
Pasadena, CA 91109

Multispectral imagery will be used in planetary exploration missions, such as a Mars Rover, to characterize surface geology. An imaging spectrometer collects images of the surface reflectance intensity at multiple discrete wavelengths (up to several hundred bands) in the visible and near infrared regions of the spectrum. The resulting spectral curve for each point in the image is characteristic of the mineral composition.

Due to limitations on communication bandwidth and time required to send data back to earth, much of the analysis and interpretation of the spectral data will be done automatically, on board the space craft. Analytic activities include recognizing and classifying spectra, and flagging those spectra which are not readily identified as known minerals. The analytic activities must be performed with limited computational power and memory, in real time.

The current system employs a hierarchy of neural networks which places spectra into progressively more detailed geologic classes, based on a selected subset of the total spectral bands. At the most detailed level of the hierarchy, a measure of classification accuracy is performed. Those spectra for which the best match is a poor match are identified, and subject to alternative processing. The goals of the "unknown spectrum" analysis are to identify both spectral and spatial feaures of interest, and avoid overlooking unidentifiable but potentially important spectra. A reasonable expectation is that many unknown spectra will derive from mixtures of known minerals, so a first step in analysis is to attempt to decompose spectra as mixtures of the closest known mineral and other minerals.

Another method of dealing with unknown spectral patterns is to cluster together all similar unknowns for transmission and human examination. Both neural network and standard clustering approaches have been considered. However, clustering on multiband data is slow. If all bands are given equal importance in a clustering procedure, unimportant bands may override the important features. Some knowledge of an area's spectral composition may be used to choose important bands for clustering.

An alternative approach attempts to limit the number of bands used for clustering, while incorporating information on the spatial distribution of the "unknown" spectra to help determine the bands of importance. Clustering on ratios of several bandsets is followed by analysis of the spatial distribution of the clusters. Bandsets which cluster to produce compact spatial regions should be of greatest interest. Iterations of alternating spectral and spatial analysis may allow the system to identify new spectral patterns with interesting spatial distributions.

Spectral Image Classification in an Alien Environment

Susan Eberlein, Gigi Yates, and Eric Majani
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

Space exploration provides vast challenges in data analysis and image interpretation. Many interesting problems in the analysis of large complex datasets may be addressed by systems that incorporate the use of neural networks for processing. This application employs neural networks for pattern classification as part of a data analysis and control unit for an autonomous roving vehicle, such as a Mars Rover. Neural networks provide advantages through robustness in the face of noisy and incomplete data, and through inherently parallel architectures for processing high dimensional data. An additional advantage may be obtained if a network can learn to distinguish unknown patterns from those for which it was trained, and to recognize similarities among new patterns.

The application problem

An autonomous vehicle for lunar or planetary exploration and sample acquisition will carry an array of complex instruments. This application concerns itself mainly with the science instruments, rather than those for navigation and vehicle control. Of particular interest is the imaging spectrometer, which senses the light reflectance intensity at many discrete wavelengths. The resulting multispectral image may be used to identify the mineral composition of the terrain (Figure 1). This information is important both for doing general survey of an alien environment and for selecting specific areas to collect samples and perform in situ experiments.

A roving vehicle faces severe constraints in computing power, memory size, transmission bandwidth and time. In most ordinary situations, a vehicle operating on Mars will need to analyze and make decisions for future activity without consulting Earth, due to the long turnaround time for Earth-Mars communication. An on-board analysis and interpretation system is particularly important for multispectral images because the images are too large to transmit, comprising up to a thousand data points for each pixel.

A system has been designed and implemented in software for automatically analyzing multispectral images, fusing data from spectral and spatial images, and making simple decisions regarding the next appropriate system activity, including reconfiguration of the instruments [1]. This system incorporates simulated neural networks at several points for pattern classification and feature detection. Currently efforts are underway to incorporate into the nets an ability to assess the accuracy of classification. In cases where a classification is poor, the system should be able to invoke alternative techniques for pattern recognition, and ideally to learn new patterns.

The role of neural networks

Several neural network architectures have proven to be robust pattern matchers, able to classify noisy and incomplete data correctly. This application has examined two classes of networks for pattern classification: standard back-propagation trained classifiers [2], and grandmother cell (matched filter) classifiers which have preset memories [3]. Initially a single network was trained by a back-propagation algorithm to place spectra into one of several geologic classes based on thirty-two input dimensions (spectral bands). The network was fully connected, feed-forward, with a single hidden layer. This network was analyzed to determine

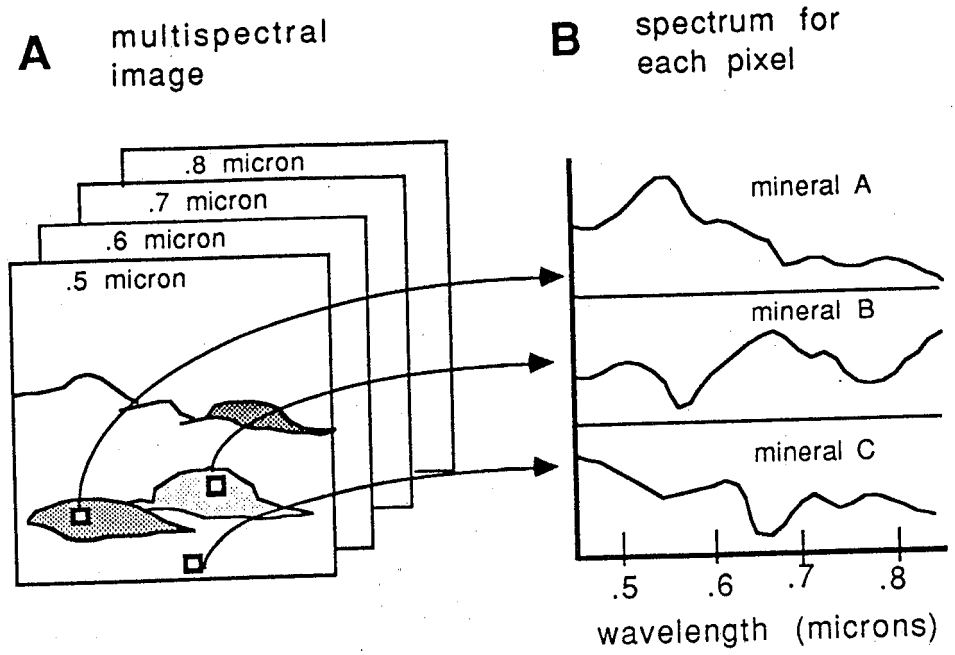


Figure 1. Multispectral images, collected from an imaging spectrometer. For each pixel, reflectance intensity values are recorded at multiple wavelengths, in narrow passbands. The resulting spectral curve may be plotted as a function of wavelength, and used to identify the composition of the material being examined, in this case, the mineral content.

which input dimensions contributed to various classification decisions, allowing a hierarchy of smaller classifiers to be developed. The smaller nets place mineral spectra into one of two broad classes based on a few spectral bands (Figure 2). The classes become progressively more detailed as a spectrum proceeds through the hierarchy.

The hierarchical approach has proven particularly efficient because it allows initial rough classification to occur with minimal computational effort, using only a few spectral dimensions. A decision making step is invoked after each level of the classification hierarchy; uninteresting regions of the image may be removed from further analysis, and instruments may be properly reconfigured for collection of further data for interesting areas.

An alternative classification approach uses grandmother cell networks to place spectra into one of several geologic classes. In this case a memory spectrum is chosen to represent each geologic class, and all thirty-two spectral bands are used as inputs (Figure 3). These nets are not trained, but the connection weights are determined in advance as a function of the normalized values of the memory vectors involved. Initially grandmother cell networks were used to place spectra into broad geological classes. They may also be employed to match a spectrum to the closest of several specific minerals within a class. In fact, the most successful combination of neural networks for this application uses the hierarchy for finding broad classes followed by a set of grandmother cell networks for final mineral identification.

Any form of pretrained neural network will only classify a pattern correctly if that pattern belongs to one of the classes used for training. In the above approaches to pattern classification, the "closest" match will be found, but that match may not represent the desired answer. Dealing with unknown or unexpected patterns is a significant issue when exploring an alien environment. It may be possible to determine before launch time what the expected distribution of most normal minerals and rocks will be on the surface of Mars, and prepare

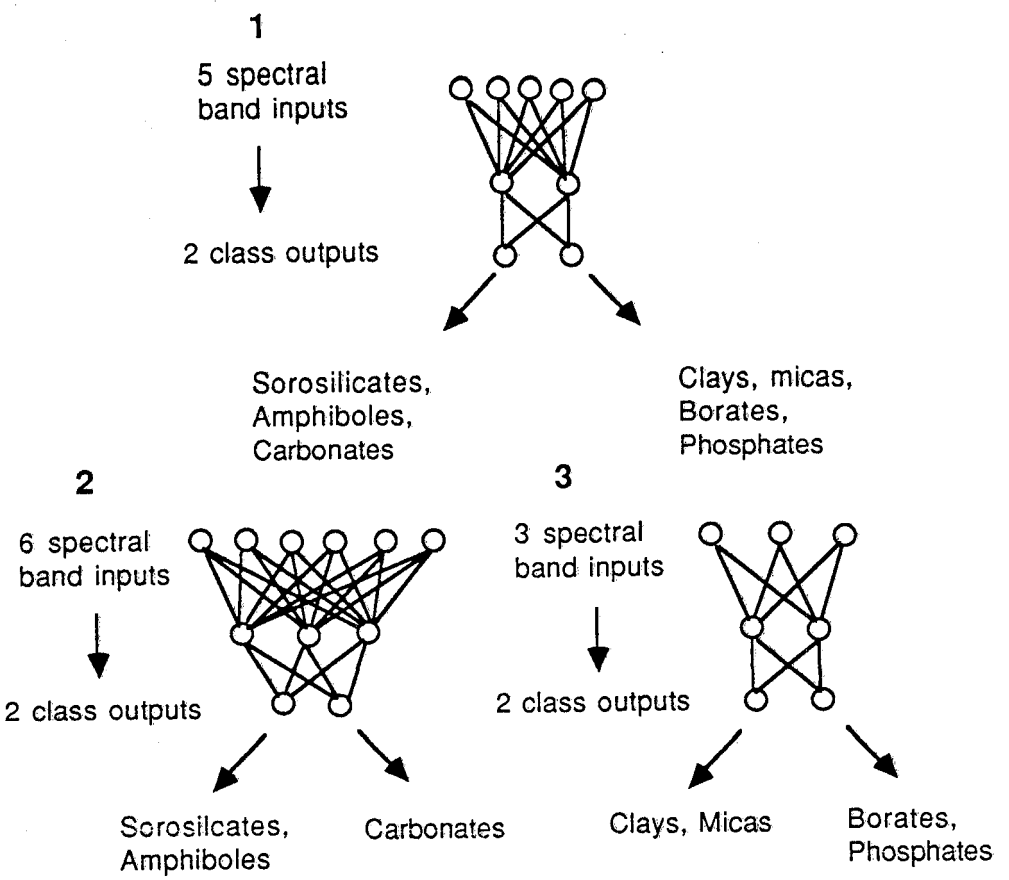


Figure 2. Hierarchy of spectral classifiers. Each classification step uses only a few spectral bands, the choice of bands depending on which geological classes are to be distinguished. Each classification step is followed by a decision on whether to continue classifying the region, based on the current scientific goal of the system. If the decision is to continue classification, the decision maker determines which spectral bands and classifier to use next.

networks to deal very efficiently with these expected cases. However, the most interesting cases will be those that are not expected, and the further a pattern is from the set of expected patterns, the more interesting it is likely to be. Thus an effective autonomous pattern classifier must recognize when a new pattern matches poorly with the expected pattern classes, and deal with the new pattern. This implies an additional requirement for the neural networks: the ability to monitor their own performance.

Networks to recognize the unknown

There are three approaches to dealing with mismatched, unknown spectral patterns.

1. One may overlook the mismatch, and simply place patterns into the closest class. This is appropriate if the training patterns or memories can reasonably be assumed to span the entire pattern space. In this case, mismatches can be attributed to noise and overlooked. However,

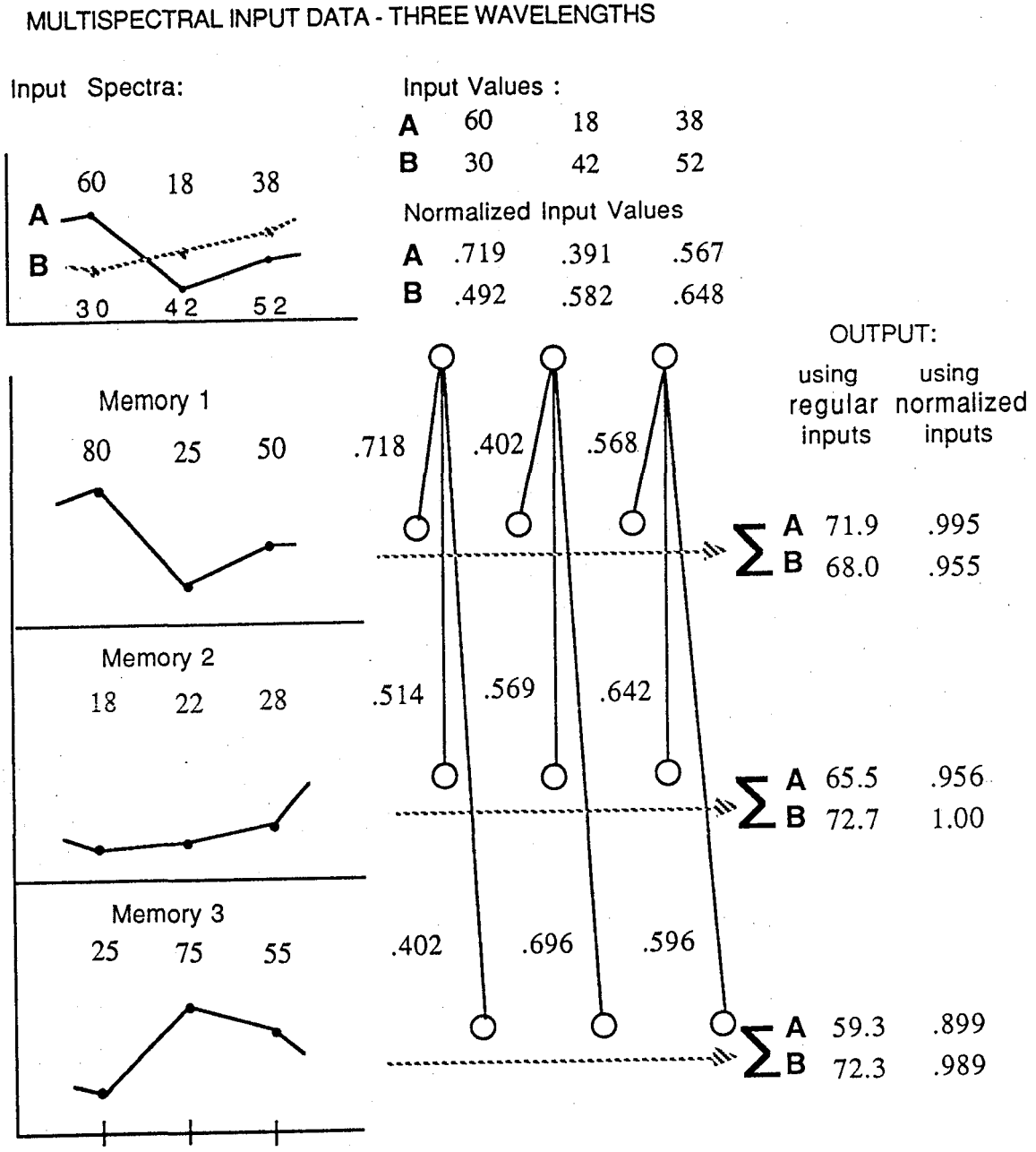


Figure 3. Classification of multispectral data with a grandmother cell pattern matcher.

Figure 3. Classification of multispectral data with a grandmother cell pattern matcher.

The grandmother cell pattern matcher performs an inner product between an input vector and each of a set of memory vectors. The memories are known mineral spectra normalized so thst the sum of squares of all components of the vector equals one. Normalization guarantees that for any given input, the inner product with the highest output value represents the best match. The input vectors consist of the intensity values at three specific wavelengths. If inputs are not normalized, the output values depend on both the spectrum shape and the mean reflectance value of the input spectrum. Mean reflectance is a function of the light shining on the scene rather than of the mineral content, so that changing the ambient lighting will produce vastly different output values for the same mineral input. When non-normalized inputs are used, the best match will still be selected, but the algorithm will not provide a measure of match accuracy or allow for comparison among several input spectra. If inputs are normalized, the highest output value for a perfect match is 1.00, allowing an assessment of how close to perfect a match is. The examples show that the numeric differences between well and poorly matched spectra are quite small. If a threshold is to be used for goodness of match (normalized inputs required), the correct choice of threshold is critical, and will vary depending on the memory being compared.

there is no guarantee that this approach will work for autonomous exploration of extra-terrestrial bodies.

2. One may find some measure for "closeness of match" so that mismatches may be rejected, then invoke alternative processing for the unknown pattern. In the exploration scenario, this probably means transmitting unexpected spectra to earth for human processing.

3. One may first identify the mismatches, then proceed to learn new categories of patterns, so that future occurrences of the same unusual patterns may be grouped together. Although this does not involve discovering anything about the identity of the unknown pattern, it does require learning the significant features of the pattern so that similar patterns may be recognized. In the remote exploration scenario, learning ability will allow the collection of information about the distribution and variation in a new spectral pattern. Then this information may be transmitted to Earth, rather than sending each single occurrence of an unknown.

The third alternative is obviously the preferred approach. It places two requirements on the neural networks used for pattern classification: they must be able to recognize poorly classified patterns and they must be able to learn new patterns. Both of these requirements are being addressed for both the back-prop trained classifiers and the grandmother cell pattern matchers.

Recognizing errors

The grandmother cell pattern matcher is designed so that euclidean distance between the input and a memory may be extracted. The memories of the network are encoded in the connection weights as the normalized values of the memory vectors (Figure 3); the sum of squares of the components for each vector equals one. If the input vectors are also normalized, the inner product between an input and a perfectly matched memory will be 1.0, while any imperfect match will be less. If the output value of an imperfect match is subtracted from 1.0, the result is a normalized version of the euclidean distance between the input vector and the memory.

In the simplistic case, one may normalize all the inputs and consider any cases where the best match is below a predetermined threshold (i.e. the distance is too great) to be incorrect. In practice there are two problems with this approach. First, in the interests of rapid processing it is undesireable to normalize all inputs in a software simulation. Normalization is unnecessary to find the closest match; it is needed only to determine an absolute measure of match accuracy. Second, a single, preset threshold for match accuracy does not work in practice, even when all input vectors are normalized. For any given mineral memory there is an expected amount of normal variability. This variability depends on the specific mineral and factors such as crystal structure and grain size. Thus a grandmother cell matcher requires a threshold for each memory, depending on the expected behavior of normal variants of that mineral.

The trained networks lend themselves to a somewhat different type of accuracy analysis. During training, the target state always has one output unit on (value = 1.0) and the rest off (value = 0.0). When training is completed and new spectra are classified, a spectrum that fits well into a known class will again produce only one active output unit. Poorer matches will cause some activity in several output units, and the amount of spurious activity is an indicator of match accuracy. Therefore a measure is needed to evaluate how confident one should be in the match provided by the highest output. Such a measure should be able to reject a match when there is significant activity in several output units, and to take into account all output units at once.

A measure that has these properties is the entropy of a probability distribution. Assume that P(i) is the probability that event i has taken place (in this case, that a given mineral has produced the spectrum in question), and that n is the total number of output units. Then the entropy:

$$H = \sum_{i=1}^{i=n} P(i) * \log\left[\frac{1}{P(i)}\right]$$

represents the overall uncertainty about the correct observance of a random event. In this case, although the values of the output units are not probabilities, they are a measure of how good a match is. Therefore if the values of the output units are normalized to add up to 1.0 (as probabilities do), and y_j is the normalized activity of output unit j, the measure:

$$H^* = \sum_{j=1}^{j=n} y_j * \log\left[\frac{1}{y_j}\right]$$

should be a reasonable measure of uncertainty about the correct match (Figure 4). If H^* is low, then the uncertainty is low, so that the confidence in a correct match is high. If H^* is high, then the match obtained is not reliable.

Learning new patterns

It is a simple matter for a grandmother cell pattern matcher to learn a new pattern. Assuming there is space available for several new memories, a new memory may be encoded by placing the normalized values of the new vector into the connections. However, this new memory does not necessarily form an accurate representation of the class of spectra being examined. Any noise in the vector has been encoded into the memory, since it is created from a single occurrence of the spectrum. Later occurrences which have different noise distribution may not be considered close matches, even though the important features are in common. This actually emphasizes a problem which already exists in the grandmother cell matching paradigm: each

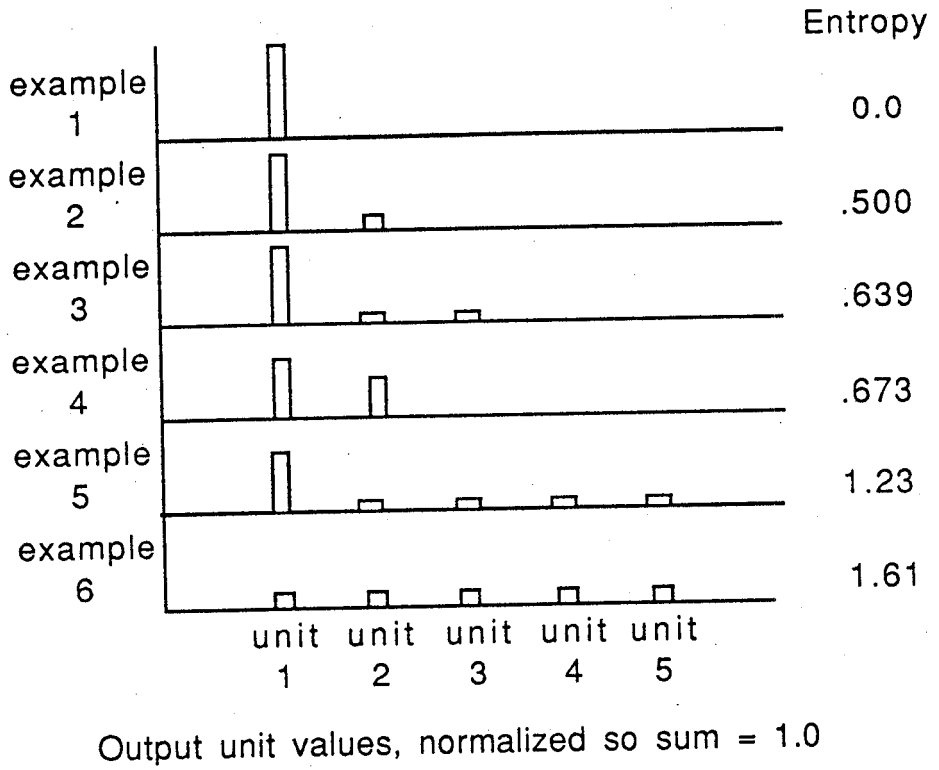


Figure 4. Illustration of entropy measure on a five-output network. The output values are normalized so the sum of all outputs equals 1. Entropy calculation (see text) represents overall uncertainty about the accuracy of a classification. Entropy (and uncertainty) is lowest when a single output is active, highest when activity is distributed over all outputs.

dimension in the memory is accorded equal importance, even though some of them may be irrelevant while others are crucial. Effective learning algorithms for the grandmother cell architecture must incorporate intelligent methods of refining a new memory until the memory encompasses the features but not the noise.

Learning new patterns in a network which was originally trained by backward propagation of an error term is more complex, but several approaches look promising. The approach under current development spawns a new output node for the network when a new class is to be learned. Unlike the grandmother cell configuration, the method for setting weights to the new output is not simple. Starting with rather arbitrary weights, unsupervised learning is required to find optimal weights.

Both grandmother cell matchers and back-prop trained classifiers have been programmed to detect misclassified patterns and attempt to learn new classes. Initial results of the efforts with the grandmother cell networks are poor, although there are several areas where improvement may be achieved. Results with the back-prop nets are much more encouraging.

Experimental results: grandmother cell networks

A grandmother cell network was designed with 10 memories, each memory consisting of a representative spectrum for a specific small geologic class. Both the memory spectra and the inputs were normalized because of the requirements for a euclidean distance accuracy assessment for the match. For each memory, an accuracy threshold was determined by classifying a

set of known spectra and monitoring the range of output distance values for those minerals known to fall into that class. The network was tested on a separate group of spectra, all of which were known to belong to one of the ten predetermined classes. In a small trial run, only 26 of 66 spectra tested by this method were classified accurately. Of the remaining 40 test spectra, 27 were recognized as being incorrectly classified by exceeding the acceptable distance threshold. The other 13 errors went undetected. In some cases, an input spectrum fell within the accuracy bounds of more than one class, due to the class specific thresholds. In this case the highest overall value was designated the best match.

The grandmother cell network was programmed to create new memories, up to a predetermined maximum number. To allow for flexibility within the framework of limited growth, a record was kept of how recently an example of a new memory had been observed. When the available space for new memories became full, those that had not been observed could be dropped to make more space. The first time an unknown spectrum was encountered, it was encoded as a new memory. If more spectra were observed which were close matches to the new memory, the memory was refined to reflect the average of these similar spectra. It was hoped that this approach would limit the problems due to noise in the new spectra. However, it was difficult to find a balance between grouping together spectra which differed only because of noise, and averaging spectra which were essentially different.

A major problem with this use of a grandmother cell classifier arises because members of a class are grouped together based on all spectral dimensions (spectral bands). Within a given geologic class, a small subset of features will be diagnostic for class recognition while most dimensions may be treated as noise. The grandmother cell architecture treats all dimensions equally. A better use of a grandmother cell network is in the identification of specific mineral spectra when the input spectrum is already known to fall into the class of interest. In this case, the input dimensions may be selected to include only those of importance for that geologic class.

A set of single class grandmother cell network have been used successfully to identify minerals after initial classification with the hierarchy of spectral classifiers (see Figure 2). Work is now underway to incorporate accuracy measures into these networks, including efforts to design accuracy measures which do not require normalizing all inputs. Since these nets are presumed to be dealing only with members of a specific geologic class, it may be possible to incorporate knowledge of the important input dimensions into the learning process. One likely type of unknown spectrum to be encountered is a mixture of known minerals (spectral reflectance values mix in an additive fashion). Use of this knowledge may also assist in devising an effective learning method.

Experimental results: back-propagation networks

The entropy measure applied to back-propagation trained networks has provided a better estimate of classification accuracy. A network was trained to place spectra into one of five geological classes, based on thirty-two input dimensions and using eight hidden units. The entropy measure was applied, with thresholds determined by the maximum entropy encountered among the correctly classified training set members. Spectra which were not part of the training set were used for testing the network. Of 115 spectra which were known to fall into one of the five output classes, 98 (85%) were correctly classified, and the remaining 17 exceeded the entropy threshold. Of 158 spectra which were known not to fall into one of the existing five classes, 129 (82%) were recognized as being poorly classified, and were designated unknown.

In an initial effort to incorporate learning, the trained network was presented with a variety of spectra, including some which did not fall into any training classes. When the entropy measure indicated a poorly classified spectrum, the net spawned a new output node. When a new output node was created, connections were made from the existing hidden units to the new output, but connections from the input to the hidden units were not altered. Values for the new connection weights were based on the hidden node output values. High hidden

node values engendered high positive connection weights, low hidden node values engendered negative connection weights. Initial results of this approach show that new nodes can be created for classifying unknown spectra, and that similar spectra encountered later may be placed into the same new class. However, considerably more work is needed on optimization of the new connection weights to guarantee that similar unknown minerals will be classified together without disrupting the classification of the known spectra.

Entropy measures are being applied to the small networks in the classification hierarchy as well. Since these networks have only two output classes based on a very few input dimensions, they may offer a simpler arena for refining the new class learning algorithms. One observation is that the number of new classes which may be learned is related to both the number of input dimensions and the number of hidden units. If a net is being designed for later expansion, its initial training must involve extra hidden units.

In general, if neural networks are to learn new pattern classes, it is critical that the important input dimensions for distinguishing the classes be recognized. This makes a grandmother cell classifier a rather poor candidate for learning new patterns, since all dimensions are treated equally. A network which has been trained to recognize classes is a better candidate, since the hidden units have come to act as feature detectors for the important input features. However, existing hidden units will only serve to recognize new combinations of features which were present in the training set. Future work will consider how much learning can be achieved for the spectral analysis problem without requiring the training of new feature detectors in the hidden layer. It may be possible to improve the likelyhood of selecting the correct dimensions for learning new classes by first using small networks to place spectra in broad classes, then performing further classification and learning based on selected spectral bands.

The problem of recognizing and dealing with unexpected patterns is of importance in many domains. In a spectral analysis system for autonomous planetary exploration, the ability to recognize unknown patterns and group together related new spectra will vastly improve the scientific return of the missions. Neural networks have good potential as tools for spectral pattern classification. Certain neural network paradigms may also prove effective for recognizing and clustering unexpected patterns. This ability will have wide ranging applications.

References

1. Eberlein, S., and G. Yates (1990). Neural network-based system for autonomous data analysis and control. In *Progress in Neural Networks,* volume 1, pp 25-55. Edited by Omid Omidvar, Ablex Publishing Corporation.

2. Rumelhart, D., G. Hinton, and R. Williams (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing,* pp 318-362. Edited by D. Rumelhart and J. McClelland, MIT Press.

3. Baum,. E.. J. Moody, and F. Wilczek (1986). Internal representations for associative memories. NSF-ITP-86-138 Institute for Theoretical Physics, University of California, Santa Barbara, California.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Machine Learning of Expert System Rules: Applications to Semiconductor Manufacturing

U.M. Fayyad

AI Group, 301-490
Jet Propulsion Lab
4800 Oak Grove Drive
Pasadena, CA 91109

K.B. Irani J. Cheng Z. Qian

Artificial Intelligence Lab
EECS Department
The University of Michigan
Ann Arbor, MI 48109

Abstract

This paper introduces a machine learning technique for aiding automated knowledge acquisition. We motivate the problem of learning classification rules from data and argue its necessity in certain industrial settings. After introducing the learning algorithm, we review some successful applications in semiconductor manufacturing. Finally we introduce two extensions to our algorithm to deal with problems encountered in industry data: noise and limited training sets.

We argue the appropriateness and necessity of machine learning to circumvent the "knowledge acquisition bottleneck", and motivate and describe the learning algorithm we developed. The GID3 system was applied to five different projects with several Semiconductor Research Corp. (SRC) industry members. We describe some of the application areas where acceptable levels of success were achieved by the program. The application areas include: identification of relationships between Reactive Ion Etching (RIE) process anomalies and the corresponding parameter settings, acquisition of a set of rules for RIE process optimization, and knowledge acquisition for an emitter piloting advisory expert system. The paper aims to bring attention to machine learning as a useful tool in the automation of the semiconductoring manufacturing process and as an aid to engineers in interpreting and assimilating experimental results.

Keywords: Machine learning, automated knowledge acquisition, decision tree induction, process diagnosis, classification.

1 Introduction

There is a steadily increasing drive towards automation in all aspects of human endeavor. Automation promises cost-effectiveness, reliability, predictability, and accuracy. So far, only fixed simple tasks in manufacturing have been automated. Automation of more complex tasks, currently requiring intelligent decision making or problem solving on the part of humans, is a much more difficult task.

One of the goals of Artificial Intelligence (AI) research is to provide mechanisms for emulating human decision-making and problem solving capabilities, using computer programs. The first AI attempts at such systems appeared as part of the technology known as "expert systems". Expert systems are intended to provide the means of encoding human knowledge about a specific task in terms of situation-action rules. The idea is that if such systems are endowed with sufficient knowledge of the task at hand, they may be able to emulate human expert behavior in most, if not all, situations that arise during task execution.

Even with such a constrained and narrow goal, serious difficulties arose that hindered the development of successful expert system applications. The first such difficulty is known as the "knowledge acquisition bottleneck" [Fieg81]. Human experts find it difficult to express their knowledge, or explain their actions, in terms of concise situation-action rules. If pressed to do so, they typically produce rules that are incorrect, or that have many exceptions. The articulation of specific intuitive knowledge into deterministic rules is a difficult, sometimes unrealistic, problem for human experts. Interviewing domain experts to extract such knowledge is also an expensive process demanding time from experts and knowledge engineers. In addition, it is a difficult and often frustrating process for the domain experts involved. Industrial diagnostic expert systems typically require a long development time.

A second problem arises in a different situation: What if a task is not well-understood, even by the experts in that area? An example of this situation is manifested in our experience with the automation of the reactive ion etching (RIE) process in semiconductor manufacturing. We discuss some of the details of this application later in the paper. In such domains, abundant data are available from the experiments conducted, or items produced. However, models that relate how output variables are affected by changes in the controlling variables are not available. Experts strongly rely on familiarity with the data and on "intuitive" knowledge of such a domain. How would one go about constructing an expert system for such a domain?

1.1 The Machine Learning Approach

The machine learning approach to circumventing the aforementioned hurdles calls for extracting classification rules from data directly. Rather than require that a domain expert provide domain knowledge, the learning algorithm attempts to discover, or induce, rules that emulate expert decisions in different circumstances by observing examples of expert tasks.

A training example consists of a description of a situation and the action performed by the expert in that situation. The situation is described in terms of a set of *attributes*. An attribute may be *continuous* (numerical) or *discrete* (nominal). For example, a nominal attribute may be *shape* with values { *square, triangle, circle*}. An example of a continuous attribute is pressure or temperature. The action associated with the situation, the *class* to which the example belongs, is a specification of one of a fixed set of allowed actions. The class of each training example is typically determined by a human expert during normal execution of his/her task. Example actions may be *raise temperature, decrease pressure, accept batch,...* The goal of the learning program is to derive conditions, expressed in terms of the attributes, that are predictive of the classes. Such rules may then be used by an expert system to classify future examples. Of course, the quality of the rules depends on the validity of the conditions chosen to predict each action.

A training example is therefore a list of the values of all the attributes along with the class to which the example belongs. Assume there are m attributes $A_1, \ldots, A_m$, p classes $C_1, \ldots, C_p$. A training example is

example	Selectivity	Δ line width	class
e-1	normal	normal	*power is high*
e-2	normal	high	*power is low*
e-3	high	high	*power is low*
e-4	high	low	*power is high*
e-5	low	normal	*flow rate is low*
e-6	low	high	*flow rate is low*

Table 1: A Simple Training Set of Examples.

an $m+1$-tuple $\langle b_1, b_2, \ldots, b_m; C_j \rangle$, where each b_i is one of the values of the attribute A_i: $\{a_{i1}, \ldots, a_{ir_i}\}$, and C_j is one of the p classes. A rule for predicting some class C_j consists of a specification of the values of one or more attributes on the left hand side and that class on the right hand side.

As an example, consider the simplified small example set shown in Table 1. This set consists of six examples e-1 through e-6. There are two attributes: *selectivity* and Δ *line width.* The attributes can take the values *low, normal,* and *high.* There are three classes: *flow rate is high, power is low,* and *power is high.* A simple rule consistent with these examples may be:

IF (Selectivity = low) THEN *Flow rate is low*

Note that this is only an illustrative simplification. Typically, the number of examples of a meaningful training set is at least in the hundreds, while the number of attributes is usually in the tens.

1.2 Further Motivation for Machine Learning

In addition to the motivations listed above, two other reasons exist for the need of a machine learning approach. The first is the growing number of large databases that store instances of diagnostic tasks. Such data is typically accessed by keyword or condition lookup. As the size of the database grows, such an approach becomes less effective. Suppose an expert needs to look up cases similar to a case being diagnosed. A query may easily return hundreds of matches. A method for determining relevant conditions automatically would be needed in this case.

Another motivation is the evolution of complex systems that have an error detection capability. Communication networks are an example. Faults are detectable by the network hardware. Several thousand faults may occur during a day. To debug such a network, a human would need to sift through large amounts of data in search of an explanation. An automated capability of deriving conditions under which certain faults occur may be of great help to the engineer in uncovering underlying problems in the hardware.

Both of the above situations indicate that machine classification learning is a potentially powerful method for summarizing large amounts of data effectively.

1.3 Industrial Applications and Problems

There are several approaches to inducing diagnostic rules from data. In this paper we do not cover all the details, nor do we review the relevant machine learning literature. We restrict our discussion to briefly presenting the problem and its complexity, and then we focus our attention on the induction of decision trees as an efficient solution. We illustrate this discussion with simple examples. We then briefly motivate and outline our algorithm (GID3) for inducing decision trees. The second part of the paper provides some details of several industrial applications in semiconductor manufacturing domains for which GID3 was utilized and was found useful by the process and knowledge engineers.

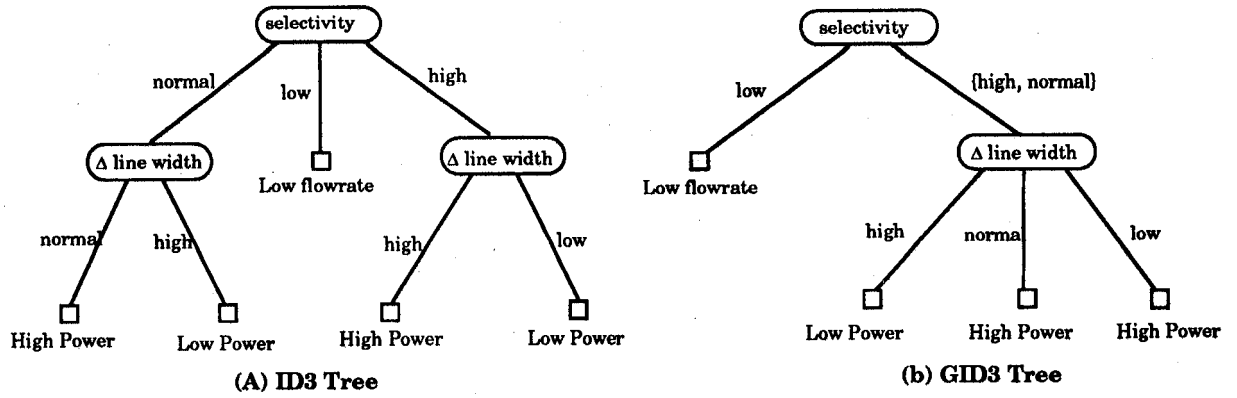


Figure 1: Decision Trees Generated for the Simple Data Set of Table 1.

We also cover two problems that we faced in our dealings with industrial data and the schemes we developed to overcome them. The typical assumption is that large amounts of data are available when machine learning is to be applied. However, there are cases when experiments may be very expensive. In such cases, training data are limited. We developed a system, KARSM, that uses Response Surface Methodology, coupled with GID3, to generate rules under such conditions. Another problem we face with industrial data is that in some processes the data may be noisy. Human recording errors, limited sensor resolution, or sensor or equipment reliability problems introduce inaccuracies in the values of the attributes. We developed a system, RIST, that utilizes statistically robust techniques along with GID3 to deal with the noise problem. Both of these systems will be briefly outlined later in the paper.

2 Inducing Rules from Training Examples

Assume that there are m attributes as described above. Let each attribute A_i take on one of r_i values $\{a_{i1}, \ldots, a_{ir_i}\}$. Assuming that on average an attribute takes on one of r values, there are $p \cdot (r+1)^m$ possible rules for predicting the p classes. It is computationally infeasible for a program to explore the space of all possible classification rules. In general, the problem of determining the minimal set of rules that cover a training set is NP-hard. It is therefore likely that a heuristic solution to the problem is the only feasible one. A particularly efficient method for extracting rules from data is to generate a decision tree [Brie84, Quin86]. A decision tree consists of nodes that are tests on the attributes. The outgoing branches of a node correspond to all the possible outcomes of the test at the node. The examples at a node in the tree are thus *partitioned* along the branches and each child node gets its corresponding subset of examples. A popular algorithm for generating decision trees is Quinlan's ID3 [Quin86], now commercially available.

ID3 starts by placing all the training examples at the root node of the tree. An attribute is then chosen to partition the data. For each value of the chosen attribute, a branch is created and the corresponding subset of examples that have the attribute value specified by the branch are moved to the newly created child node. The algorithm is then applied recursively to each child node until either all examples at a node are of one class, or all the examples at that node have the same values for all the attributes. An example decision tree generated by ID3 for the sample data set given in Table 1 is shown in Figure 1(a).

Every leaf in the decision tree represents a classification rule. The path from the root of the tree to a leaf determines the conditions of the corresponding rule. The class at the leaf represents the rule's action.

Note that the critical decision in such a top-down decision tree generation algorithm is the choice of

attribute at a node. The attribute selection is based on minimizing an information entropy measure applied to the examples at a node. The measure favors attributes that result in partitioning the data into subsets that have low class entropy. A subset of data has low class entropy when the majority of examples in it belong to a single class. The algorithm basically chooses the attribute that provides the locally maximum degree of discrimination between classes.

3 Overcoming Problems with ID3 Trees

It is beyond the scope of this paper to discuss the details of the ID3 algorithm and the criterion used to select the next attribute to branch on. The criterion for choosing the attribute clearly determines whether a "good" or "bad" tree is generated by the algorithm. Since making the optimal choice of attribute is computationally infeasible, ID3 utilizes a heuristic criterion which favors the attribute that results in the partition having the least information entropy with respect to the classes. This is generally a good criterion and often results in relatively good choices. However, there are weaknesses inherent in the ID3 algorithm that are due mainly to the fact that it creates a branch for each value of the attribute chosen for branching.

3.1 Problems with ID3 Trees

Let an attribute A, with possible values $\{a_1, a_2, \ldots, a_r\}$ be chosen for branching. ID3 will partition the data along r branches each representing one of the values of A. However, it might be the case that only values a_1 and a_2 are of relevance to the classification task while the rest of the values may not have any special predictive value for the classes. These extra branches are harmful in three ways:

1. **They result in rules that are overspecialized.** The leaf nodes that are the descendants of the nodes created by the extraneous branches will be conditioned on particular irrelevant attribute values. Since each leaf node corresponds to a classification rule, the irrelevant conditions will appear in the preconditions of the corresponding rules.

2. **They unnecessarily partition the data,** thus reducing the number of examples at each child node. The subsequent attribute choices made at such child nodes will be based on an unjustifiably reduced subset of data. The quality of such choices is thus unnecessarily reduced.

3. **They increase the likelihood of occurence of the** *missing branches problem.* This problem occurs because not every possible combination of attribute values is present in the examples.

The third problem can be illustrated in the ID3 tree shown in Figure 1(a). Consider two possible unclassified examples which are to be classified by the tree of Figure 1(a):

ex1: (Selectivity = low) & (Δ line width = low)
ex2: (Selectivity = normal) & (Δ line width = low)

Both ex1 and ex2 are examples that have combinations of attribute values that did not appear in the training set of Table 1. However, the tree readily classifies ex1 as being the result of an etch where the *flow rate was low*, but ex2 fails to be classified by the tree. This is because the subtree under the normal selectivity branch has no branch for *low Δline width.* We shall shortly illustrate how the occurence of *missing branches* may be avoided.

The main problem with the tree of Figure 1(a) is that the normal and high selectivity branches should not be separated. Low selectivity is the only value of relevance to the occurence of a problem. It would be desirable if the learning algorithm could somehow take account of such situations by avoiding branching on

attribute values that are not individually relevant. This would reduce the occurence of the three problems listed above.

3.2 The GID3 Algorithm

To avoid some of the problems described above, we developed the GID3 algorithm[1]. We generalized the ID3 algorithm so that it does not necessarily branch on each value of the chosen attribute. GID3 can branch on arbitrary individual values of an attribute and "lump" the rest of the values in a single *default branch.* Unlike the other branches of the tree which represent a single value, the default branch represents a subset of values of an attribute. Unnecessary subdivision of the data may thus be reduced. Figure 1(b) shows the tree GID3 would generate for the data set of Table 1. Note that both examples, ex1 and ex2, are classifiable by this tree. The missing branch problem that prevented the tree of Figure 1(a) from classifying ex2 does not occur in the tree of Figure 1(b).

We now turn our attention to the application of GID3 to problems in the domain of semiconductor manufacturing. Interested readers are referred to [Fayy91, Chen88] for detailed accounts of the ID3 and GID3 algorithms, the attribute selection criterion, the weaknesses of the ID3 approach, and various performance measures to evaluate the quality of the resulting trees. Performance measures include error rate in classifying new examples and measures of the size complexity of the generated tree.

4 Applications of GID3 in Semiconductor Manufacturing

In this section we discuss several applications of the GID3 algorithm to semiconductor manufacturing domains. Most of these domains involve the reactive ion etching (RIE) process. The RIE process is a wafer etching process that promises increased precision and higher device density. It has been targeted for automation by the Semiconductor Research Corporation (SRC), a consortium of major U.S. companies in semiconductor manufacturing. One of the steps necessary for automation is the development of expert systems that determine process parameter settings based on given output constraints. The problem is that the process is not well-understood and no satisfactory methods for determining proper control settings exist.

To illustrate the types of industrial tasks to which GID3 can be applied, we describe application tasks from three categories: RIE process diagnosis, RIE process optimization, and an emitter piloting application. The goal of process diagnosis is to derive rules for diagnosing faults by deciding which process parameters are not correctly set. We describe two projects in this category. In the process optimization category, the project targeted deriving rules for dealing with situations where the operating point drifts away from the optimal operating point in the parameter space. Finally, we briefly discuss our effort on a project aimed at facilitating the knowledge acquisition effort for the development of an emitter piloting advisory expert system.

4.1 Process Diagnosis

The first project's goal is to acquire a set of RIE process diagnostic rules from a collection of production log data that contain fault inspection results by process engineers. The goal is to etch a specified pattern in the metal on a wafer.

Each data log contains about 60 data entries, including machine type, device specification, material and resist thickness, plasma time, power, DC bias, chamber pressure, gas flow, temperature, valve position and number of wafers. Since this is a multi-stage (three stage) etch, each stage has its own set of measurements. A distinct table slot is used for visual inspection after the etch. For this project, three types of inspection

[1]The name GID3 stands for Generalized ID3.

power1	power2	power3	time1	time2	time3	wafer#	topology	...	outcome
1117	1134	490	8.98	7.20	5.0	18	v	...	erosion
895	1139	492	10.26	8.2	5.0	24	v	...	erosion
833	854	442	7.4	6.0	5.0	1	v	...	sleeves
835	818	491	9.7	7.76	5.0	5	v	...	none
859	835	490	9.9	7.92	5.0	2	v	...	none
867	886	466	9.9	7.9	5.0	7	v	...	sleeves
847	871	473	9.8	7.8	5.0	8	v	...	sleeves
776	833	500	8.6	6.9	5.0	8	v	...	none
771	813	490	8.7	7.0	5.0	4	v	...	none
847	825	491	9.20	7.36	5.0	13	v	...	erosion
851	843	490	17.2	4.3	5.0	6	v	...	none
806	896	493	10.5	8.40	5.0	2	v	...	none
844	822	493	9.16	7.33	5.0	14	v	...	erosion
860	809	490	9.8	7.84	5.0	2	v	...	none
867	825	452	9.7	7.76	5.0	2	v	...	sleeves
878	819	454	24.1	14.4	5.0	1	t	...	none
848	816	455	321.0	16.8	5.	9	t	...	none
806	778	484	22.5	9.0	5.0	8	t	...	sleeves
881	795	467	22.7	13.6	5.0	1	t	...	none
772	868	490	8.7	7.0	5.0	7	v	...	none
...	...	...	...	...	...	...	...	...	...
842	826	494	17.0	13.6	5.0	1	v	...	none

Table 2: A Partial list of Data Logs for a RIE Process

results are commonplace, namely, *normal, PR erosion,* and *sleeves.* A partial list of such data logs is shown in Table 2.

Process fault diagnosis has been a regular demand on process engineers. Whenever a fault is detected, its immediate cause needs to be identified and a decision is then made to correct the problem by adjusting process parameters directly or indirectly. Determining such adjustments is a nontrivial task. Abstracting these daily routines into rules that cover the task has been found to be difficult. Extracting general rules that can be transferred across different processes and be used to guide further reasoning to find physical laws governing the observed phenomena has been especially difficult. For this project, the difficulty in diagnosing the faults is due to the large number of process parameters, most of which vary greatly. These conditions naturally make the domain appear promising for the application of machine learning techniques.

Once the attributes and classes are determined, GID3 can be used to induce a decision tree from the available data. The decision tree is then transformed into a set of diagnostic rules because the rule format is more general and is easier for process engineers to comprehend. In order to get a set of rules which are general and reliable, and to overcome problems with noisy data, we used the RIST package described later. RIST runs GID3 many times. Each time, a random subset of the given data set is selected to induce rules. The rules are then tested against the whole data set. A set of statistical criteria are used to select a subset of the rules obtained from each GID3 session. We describe this method further in section 5.1.

Each of the rules generated provides a range of operating conditions with a prediction of the process outcome under those conditions. The power of our machine learning approach is manifested by the fact that most rules can predict the process outcome correctly with only one to three tests on the attributes. Figure 2, shows four of the rules generated for the data of Table 2. These rules exemplify pieces of compact, previously unknown, knowledge that are found useful by both process and knowledge engineers.

The second project was aimed at identifying relationships between RIE process problems, such as reduction in yield, and corresponding process parameters including the flow rate of each gas component

Rule 1: IF erosion is observed
THEN probably too many wafers are loaded;
Reduce number of wafers to < 11.

Rule 2: IF erosion is observed
THEN power at etch stage II may be too high;
Reduce power at etch stage II to < 956.5V.

Rule 3: IF sleeves are observed
THEN etching at stage I may not be sufficient;
Increase etch time for stage I to > 8.15 minutes.

Rule 4: IF sleeves are observed and stage III power$\in [451, 487]$
THEN etching at stage II may not be sufficient;
Increase overetch percentage for stage II.

Figure 2: Rules for Diagnosing Faults of a RIE Process

and the chamber pressure for different etching steps. This data set was derived by regression analysis which statistically identifies the geometric patterns such as length, corners and gaps responsible for the yield loss. Classes consisted of dominating defect patterns. The details of this project with Westinghouse are discussed in [Frie89]. The rules derived by GID3 were used to analyze and understand the process behavior.

4.2 Process Optimization

To achieve high yield, low defect density, and small geometry, RIE must operate at an optimal point in the input parameter space. The problem, known as process optimization, or process design, is the most important problem that a plasma engineer has to face.

The goal of process optimization is to find a set of input parameters, usually referred to as a *recipe*, such that the outputs can be optimized. For example, one may want to minimize the line width change under the condition that selectivity is higher than a certain value and uniformity is within a certain range.

A popular method to solve the process optimization problem is the Response Surface Methodology (RSM) [Berg82]. RSM is a statistical method in which the input parameters of RIE are treated as independent variables and the output parameters as response variables. For each response variable, multiple regression analysis is applied to generate a response surface to fit the data. Optimization techniques are then used to find a point that optimizes the response variables under the given constraint [Berg82]. However, RSM has its drawbacks. It is a static method in the following sense. Given a set of experimental data, a set of response surfaces can be generated and a fixed optimal point can be found. According to this optimal point, a recipe is formulated. The problem is that under a fixed recipe, the process might not be optimal because certain hidden variables which were not considered in the design process influence the process later. In other words, the response surface may drift so that the operating point may no longer be optimal. Obviously, what we need is a set of rules which tell us where to move in the parameter space to achieve the optimal output under the given constraints. This makes for a dynamic, rather than static, solution to the optimization problem.

The above analysis lead us to consider using machine learning to process optimization problems for RIE. Our objective was to generate a set of rules that tell us which input parameters should be changed, and in which direction, if the outputs are not optimal or do not satisfy the constraints. We discuss the general methodology that we applied to a particular project. Table 3 shows a partial set of experimental data we aquired from industry for this project. *Pressure*, *flow*, and *percent flow* are input parameters while the other five columns of Table 3 represent outputs. Each of the 20 items represents an experiment. To generate more data items, we used KARSM (described later in section 5.2) to obtain an excess of 500 examples from these 20.

To apply GID3, obviously, we can regard input parameters as attributes and output parameters as outcomes. Since GID3 requires that each example have one class (outcome), we first discretize and combine the output parameters. For the constraint variables, the values can be discretized as either *low* or *high* (based on the constraints). For the optimization variables, the values are discretized as different

No.	*press.* (mTorr)	*flow* (sccm)	*% flow* (%)	*Δ CD* (μm)	*CD unif.* (μm)	*Oxide loss* ($\mathring{A}$)	*Oxide unif.* ($\mathring{A}$)
1	300	150	25	0.05	0.05	368	216.0
2	500	150	25	0.25	0.05	316	43.5
3	300	300	25	0.18	0.40	407	162.0
...	...	...	...	...	...	...	
20	400	225	38	0.10	0.05	220	38.0

Table 3: Experimental Data for Process Optimization

levels such as *very high, high, medium, low, very low.* The data is then fed into GID3 and a decision tree is obtained. In the decision tree, a leaf represents a class of outputs. A leaf may represent an optimal class, for instance, ⟨*very low, high, low, low*⟩ for "very low line width change, high selectivity, and low CD and Oxide uniformity". This is an optimal class when one wants to minimize the line width change under the condition that selectivity is higher than a certain value and uniformities are lower than certain values. Other leaves may represent faulty classes. For instance: "high line width change, low selectivity, low CD uniformity, and high Oxide uniformity". To derive the required rules, the condition—the path from tree root to leaf—of a faulty leaf is compared to the condition of a leaf having an optimal class. The differences represent the changes needed to bring a faulty operation to an optimal one. For example, assume that the condition of the optimal leaf is "pressure below 300 mTorr, total flow above 180 sccm, percent flow above 40 %" and that the condition of a faulty leaf is "pressure above 312 mTorr, total flow from 215 to 300 sccm, percent flow below 37.5 %". We can now derive a rule

> "if selectivity is lower than normal, Oxide uniformity is higher than normal, line width change is high, to minimize line width change, pressure should be decreased and percent flow should be increased".

4.3 An Emitter Piloting Expert System Application

GID3 has also been applied successfully to acquire knowledge for minimizing steps in emitter piloting dispositions. Emitter piloting is a process of tuning integrated circuits printed in wafer so that device specification can be satisfied. This task is typically carried out by a human operator. The initial cycle time is determined by experience and cycle time adjustment is then guided by the measurement of two device parameters. If the values of the two parameters fall in their respective desired ranges, the cycle time is accepted for batch tuning and is called the "shooting time". Otherwise, it is either increased or decreased to bring parameter values within desirable ranges. The number of steps needed before success in such a process is greatly affected by operator experience. Such experience is very valuable but is very difficult to encode in rules. The purpose of the project was to collect all sequences of trials conducted and to use the machine learning approach to attempt to extract the knowledge underlying the actions of human operators. See [Yang89] for further details of this domain.

The raw data used in knowledge acquisition is composed of numerous experiment data logs, each of which consists of sequences of cycle time adjustments targeting one shooting time. For every trial in each sequence, the cycle time used and two parameter measurement values were recorded. GID3 was used to learn rules for jumping to a shooting time from an arbitrary cycle time by letting each data point be the condition under which certain adjustments can be made to achieve a certain shooting time. The difference between the current cycle time and the actual shooting time for each example was taken to be the predetermined class. The rules induced by GID3 were evaluated by an expert and were deemed satisfactory. In this project GID3 was used as a knowledge acquisition tool to gather rules for incorporation into an expert system developed by Harris Semiconductor.

5 Dealing with Industrial Data Problems

In conclusion to this presentation, we focus on two special problems encountered in industrial applications and the solutions we devised to combat them. The problems are:

Noisy Data: attribute values may be erroneous due to human recording errors, imperfect sensor repeatabilty, or defects in process equipment or sensors.

Limited Training Data: the training data may be small in size and conducting more experiments may be too costly.

5.1 Dealing with Noisy Data

Empirical learning algorithms are typically sensitive to the presence of noise because they rely solely on data to discover rules. Typically, they are not intended to have access to special domain knowledge to guide their decisions. Noise in a training data set may cause irrelevant rule conditions to be selected. The solution we devised combats noise in two ways:

1. Statistical evaluation (pruning) to identify and remove irrelevant conditions from rules.
2. Random sampling of multiple training sets and selection of statistically significant rules from the trees generated for these training sets.

After a decision tree is generated, a statistical test is applied to remove rule preconditions that are deemed statistically irrelevant, resulting in more general, *pruned*, rules. The statistical test we use is Fisher's Exact Test. It measures the probability that a hypotheses (in this case a condition of a rule) is irrelevant to the outcome. If this probability is higher than a small value, the condition is discarded. For details of the statistical test see [Quin87, Finn63].

This method of statistical pruning deals with the rules produced from one run of GID3. By randomly sampling subsets from a given training set, many trees can be generated. For each tree, we apply the statistical test and keep only the "good" rules. Finally, a subset of the surviving rules that covers the original training set is selected. When coupled with a method for statistical significance testing, the multiple random sampling of the training set has proven to be an effective technique for extracting a compact and reliable set of rules from the original training set. The method, illustrated in Figure 3, has been implemented in a software package named RIST (Rule Induction and Statistical Testing).

5.2 Dealing with Limited Training Sets

The learning algorithms we discussed do not use any special domain knowledge about the data during tree or rule generation. They rely on the availability of large training samples to detect the presence of meaningful reliable patterns or correlations. In some cases however, obtaining training examples may be an expensive process. In this case, only a limited training set may be available.

In section 4.2 we mentioned that RSM is standard methodology used in process optimization. The core component of this methodology is to approximate the given data by a polynomial surface. Once the data is fitted with a surface, each point on the surface corresponds to a specification of the input and output variables. Since we know the target conditions that the output variable must satisfy, we can quantize the output value into: { *Good, Bad*}, or into a finer partition such as { *Very Low, Low, Good, High, Very High*}.

At this point, we can randomly choose a combination of input values, and the response surface will give us the associated discretized output. As is shown in Table 4, 500 examples can be extracted from the original data of Table 3. This gives us a method of extracting an arbitrarily large training sample from an

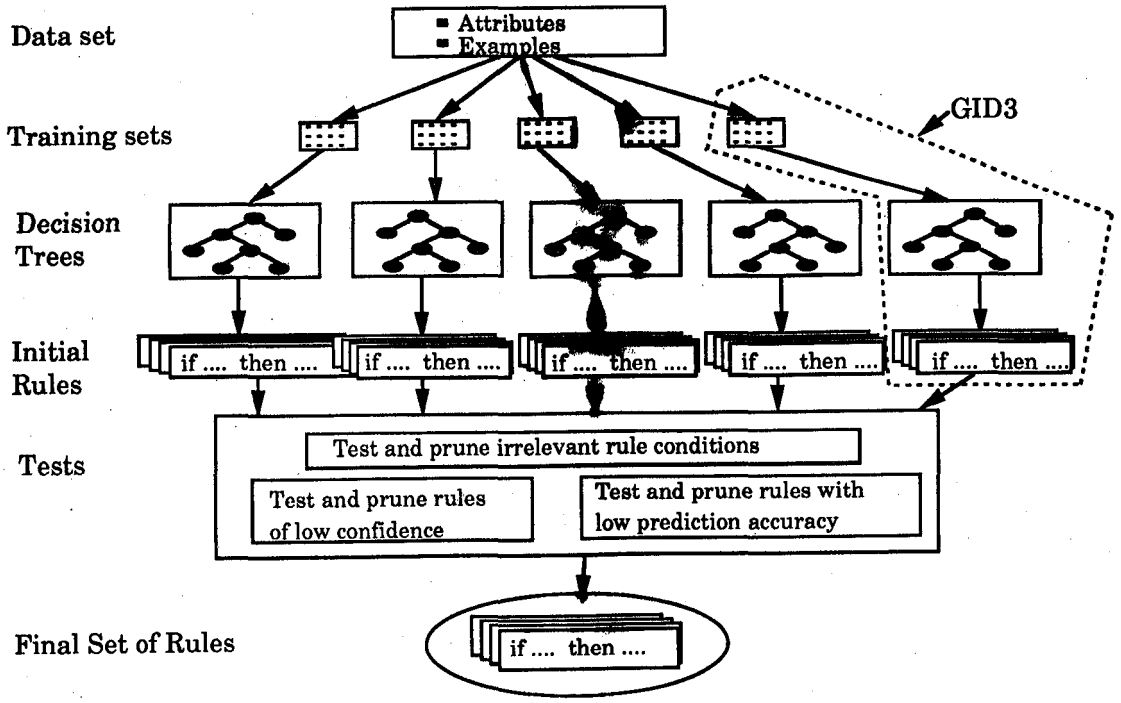


Figure 3: Data Flow Diagram for RIST

original small training set. Of course, the "usefulness" of the extracted large training set depends on the correctness of the response surface generated to fit the original data. The generated data set is then fed into RIST (or GID3) to extract qualitative rules that describe the behavior of the data.

We have implemented this procedure in a software package called KARSM (Knowledge Acquisition from Response Surface Methodology). It is Illustrated in Figure 4.

The reader may wonder why we do not stop with the response surface and use it directly to describe the process. The answer is that the response surface does not give an easily *understandable* description of the process. By generating rules from the surface we essentially extract a *qualitative* description of the behavior of the process in terms of desirable and undesirable regions of the output space. The procedure for extracting, from the decision tree, the qualitative rules that specify the direction in which process parameters should be changed, when the operating point drifts, is described in section 4.2.

No.	*pressure*	*total flow*	*percent Cl_2 flow*	*class*
1	377	192	49	ab,l,l,l
2	342	297	36	h,l,l,h
3	491	241	34	ab,h,l,h
...	...	...	...	...
499	452	211	37	m,h,h,l
500	303	185	42	vl,h,l,l

Table 4: Classified Random Samples

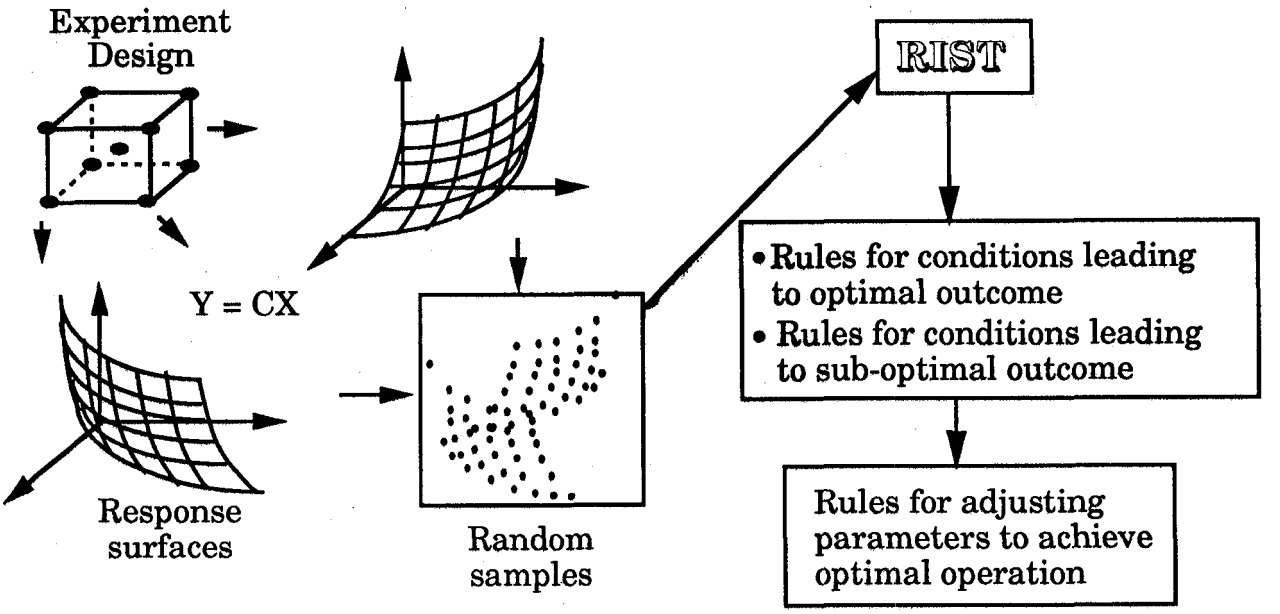


Figure 4: Data flow diagram for KARSM

6 Conclusion

We have introduced the general problem of extracting rules from data for the purpose of automating the knowledge acquisition process. We described the GID3 algorithm for inducing decision trees and its use in automating knowledge acquisition in several application domains. Given a training set of data, the algorithm produces a decision tree for predicting the outcome of future experiments under various, more general conditions. The tree may then be translated into a set of rules for use in expert systems. We also introduced two extensions of GID3 to deal with noise and limited training set availability.

We have utilized GID3 to extract knowledge from data in several application domains. In each case, the decision trees obtained, or the English version of the generated rules, were sent to the engineer who supervises the experiments and provides the data. The derived rules were judged to be consistent with the data and conformant with the engineers' expectations. A derived model, or a discovered pattern, that is consistent with a process engineer's expectation is of great value in two ways:

1. It provides a previously unavailable mechanical means for classifying events or relating faults to parameters.

2. It gives the process engineer further insight into the process by making explicit a pattern that was previously implicit as part of the engineer's "intuition" about the process.

We believe that machine learning techniques have an important role to play in the automation and improvement of manufacturing techniques as well as many other diagnostic tasks. A machine learning approach avoids the ubiquitous "knowledge acquisition bottleneck" by minimizing the required interaction with domain experts and focussing on a resource that is much easier to obtain: experimental data. A further advantage of using an induction system such as GID3 is that knowledge acquisition can be fast, accurate, and most importantly, can be automated.

Acknowledgements

This work was supported by the University of Michigan SRC Research Program under contract #89-MC-085. We would also like to thank Hughes Microelectronics Center for their partial support of this work in the form of an unrestricted grant.

References

[Berg82] Bergendahl, A.S., Bergeron, S.F., and Harmon, D.L. (1982). "Optimization of plasma processing for silicone-gate FET manufacturing applications." *IBM Journal of Res. Dev. vol. 26*, no. 5.

[Brie84] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and Regression Trees.* Monterey, CA: Wadsworth & Brooks.

[Chen88] Cheng, J., Fayyad, U.M., Irani, K.B., and Qian, Z. (1988). "Improved decision trees: A generalized version of ID3." *Proceedings of the Fifth International Conference on Machine Learning.* pp. 100-108. Ann Arbor, MI.

[Fayy91] Fayyad, U.M. (1991). *On the Induction of Decision Trees for Multiple Concept Learning.* Dissertation in preparation, EECS Dept., The University of Michigan, Ann Arbor.

[Fieg81] Fiegenbaum, E.A. (1981). "Expert systems in the 1980s." In Bond, A. (Ed.), *State of The Art Report on Machine Intelligence.* Maidenhead: Pergamon-Infotech.

[Finn63] Finney, D.J., Latscha, R., Bennett, B.M., and Hsu, P. (1963). *Tables for Testing Significance in a 2x2 Contingency Table.* Cambridge: Cambridge University Press.

[Frie89] Friedhoff, C.B., Cresswell, M.W., Lowry, C.R., and Irani, K.B. (1989). "Analysis of intra-level isolation test structure data by multiple regression to facilitate rule identification for diagnostic expert systems." *Proceedings of the International Conference on Microelectronic Test Structures.* Edinburgh, Scotland.

[Quin86] Quinlan, J.R. (1986). "Induction of decision trees." *Machine Learning 1, No. 1.* pp. 81-106.

[Quin87] Quinlan, J. R. (1987). "Generating Production Rules From Decision Trees." *Proceedings of the Tenth International Joint Conference on Artificial Intelligence.* Milan, Italy.

[Yang89] Yang, Y.K. (1989). "EPAS: An emitter piloting advisory expert system for IC emitter disposition." *Proceedings of Semicon West.*

Inferring Causal Structure In Mixed Populations

by Clark Glymour, Peter Spirtes, and Richard Scheines

1. Introduction

In this paper we will examine the problem of reliably inferring causal relations from statistical data and fragmentary background knowledge. Such causal inference problems arise in many instances in statistics, sociology, economics, and epidemiology, among others. The problem can also arise when building expert systems that use Bayes networks. In many cases such networks are constructed on the basis of some expert's background knowledge; in many other cases however our background knowledge is woefully inadequate for constructing a useful expert system.

The causal structure among a set V of n random variables can be represented by a directed graph over V, where there is an edge from A to B if and only if A is a direct cause of B relative to V. (We will say that A is a direct cause of B relative to V if and only there is a causal chain from A to B that does not involve any of the other variables in V.)

Given a joint distribution over V, the sheer number of different possible causal theories over V makes inferring which causal structure generated the joint distribution extremely difficult. There are $\binom{n}{2}$ pairs of variables in V, and for each pair of variables there are four possibilities: A causes B, B causes A, neither causes the other, or both cause the other (which we interpret as a feedback loop). Hence, there are $4^{\binom{n}{2}}$ different causal structures over V. If n = 6, there are 1073741824 different theories; if background knowledge eliminates the possibility of cycles, there are approximately 3000000 theories; and if background knowledge provides the time order for each pair of variables, then there are approximately 32768 different theories. For 12 variables, the corresponding numbers are 5444517870735015415413993718908291383296, 52193965134382940502050406 3, and 73786976294838206464. It is not uncommon for medical models or econometric models to include several hundred variables.

We will describe an algorithm that efficiently and reliably infers causal structures from statistical data (under the assumption that every common cause of a pair of measured variables is itself measured.) In order to execute the algorithm (called PC), it is necessary to determine for certain pairs of variables a and b, and certain sets of variables C, whether a and b are independent

conditional on C (in the discrete case), or whether the partial correlation $\rho_{ab.C}$ vanishes (in the linear case.) (This is also true of a number of other causal inference algorithms.)

We will then examine the extra difficulties that are posed by populations that consist of mixtures of sub-populations in which each sub-population has the same causal structure, but the strength of the causal connections differ. In these populations, conditional independence relations that hold in each sub-population generally do not hold in the population as a whole. Also, partial correlations that vanish in each sub-population generally do not vanish in the population as a whole. In such mixed populations, PC cannot be reliably employed.

Finally, we will show in the special case that the variables are linearly related, each unit in the population has the same causal structure, but the linear coefficients are independently distributed, the partial correlations are the same as those in a population generated by the same causal structure in which the linear coefficients do not vary. In populations in which the linear coefficients are independently distributed, PC can be reliably employed.

2. Pseudo-Indeterministic and Indeterministic Systems

Consider any collection of finite structures in which there is a set of input variables whose values are independently and randomly distributed, a set of variables each of which is some function--linear, non-linear, or whatever--of some subset of the input variables, a set of variables each of which is some function of some subset of the union of set of input variables and the set of first-level variables, and so on. We call such a collection of random variables, functional relations, and distributions over the independent variables a **causal system**.

In all of these cases there is a common formal structure. The causal structure can always be represented by a *directed graph*. In circuits without feedback and in most applied statistical cases, the directed graph is acyclic. Henceforth we will consider only acyclic directed graphs. Fig. 1 is an example of a graph of a causal structure.

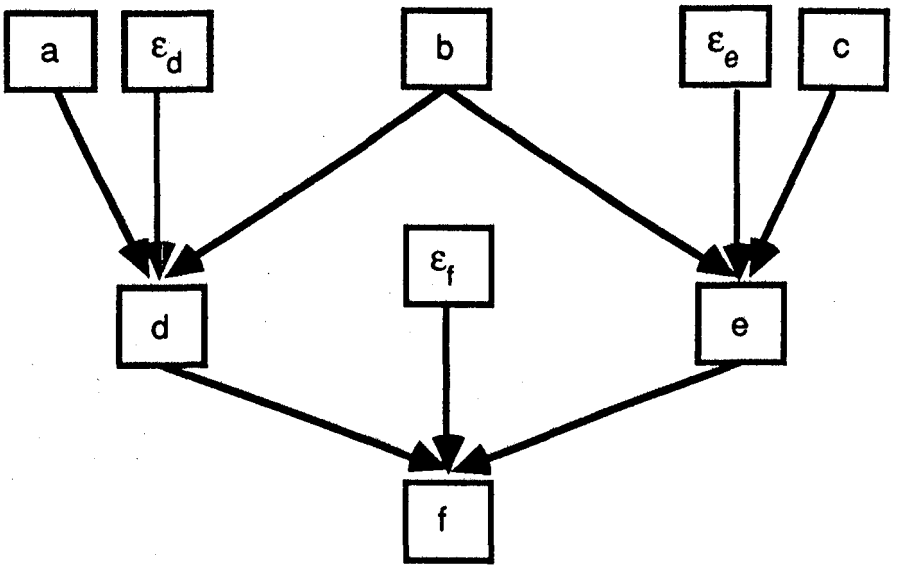


Fig. 1: Causal Structure I

Any directed acyclic graph represents a causal structure, the values of the variables represented as vertices of zero indegree (the inputs or exogenous variables) take their values randomly and independently of each other. (We assume that for any three disjoint sets A, B, and C of exogenous random variables that A is independent of B conditional on C.) Assume for the moment that the value of each variable that is not an input variable is uniquely determined by its immediate causes. Then all variables are in fact random variables, and a graph of causal relations also represents a system of functional relations among these random variables: e.g.,

a input

b input

c input

ε_d input

ε_e input

ε_f input

$d = r(a,b,\varepsilon_d)$

$e = s(b, c,\varepsilon_e)$

$f = t(d, e,\varepsilon_f)$

for any three disjoint subsets X, Y, Z of $\{a, b, c, \varepsilon_d, \varepsilon_e, \varepsilon_f\}$ X is independent of Y conditional on Z

The graph specifies that the exogenous variables are independent of each other, and for each variable the graph determines what other variables it is a function of, but the graph does not

further specify the function. We say that the graph represents the **causal structure**. Many causal systems share the same causal structure and can be represented by the same graph.

A causal system S generates a probability distribution P in the following way. Since each non-exogenous (endogenous) variable is a function of the exogenous variables (or a function of variables that are themselves ultimately functions of the exogenous variables), once values for each of the exogenous variables have been specified, the values of all of the variables in the system have been completely determined. Similarly, specifying a probability distribution over the exogenous variables completely determines a joint probability distribution over all of the variables. We will extend this terminology to causal structures as well as causal systems: if causal system S generates distribution P, and S has causal structure C, we will also say that C generated P. Each causal system S generates a unique probability distribution. However, since many different causal systems share the same causal structure, many different distributions can be generated by one causal structure.

The connection between causality, directed acyclic graphs and probability distributions described above is tacitly assumed in many of the usual causal modeling formalisms in applied statistics, e.g., in factor analysis, in linear structural equation models and in causal models with discrete variables. "Recursive" structural equation models, for example, specify a system of linear equations that can be viewed individually as regression equations with random regressors with non-zero variances. At least implicitly, there is a regression equation of this kind for each variable in the system. A system of such equations determines a directed acyclic graph, G, with its variables (omitting the error variables) as vertices in the obvious way. A joint probability distribution is imposed consistent with these assumptions.

In the complete set of variables, the value of an endogenous variable is always completely determined by its causal parents. However, in the example depicted in Fig. 1, if only the subset of variables V = {a, b, c, d, e, f} is considered, the set of immediate causes in V of each endogenous variable does not uniquely determine the value of the variable. If we assume that for each endogenous variable X in the set V there is a unique exogenous variable of unit outdegree and non-zero variance (an "error" variable) not in V, that together with the causal parents of X in V completely determines the value of X, we call the causal structure of such a set of variables **pseudo-indeterministic**. The distribution generated by a pseudo-indeterministic causal system is the marginal of a distribution generated by a deterministic causal system. A particular kind of pseudo-indeterministic causal system is a **linear causal systems**, in which all of the

functions relating the variables are linear. In that case we assume that all of the variances conditional on any set of variables not including error variables are non-zero, and that all partial correlations among non-error variables exist.

In the case of pseudo-indeterministic causal systems, the reason that the values of the endogenous variables are not completely determined by their causal parents in V is that the "error" terms which partially determine the values of the endogenous variables are not in V. Another possible reason that the values of the endogenous variables are not completely determined by their causal parents in V is that there is a genuinely indeterministic relation between the endogenous variables and the complete set of its causal parents. If that is the case, we say that the causal system is **indeterministic**.

3. Causal Graphs and Bayes Networks

A Bayes network is a graph G and a distribution P that satisfies the following conditions:

> **Markov Boundary Condition:** In P, each variable in G is independent of all of its non-parental non-descendants conditional on its parents, and no proper subset of its parents satisfies this condition.

We will assume that any distribution P generated by a causal structure G satisfies the Markov Boundary Condition with respect to P.

If every conditional independence relation that holds in P is entailed by satisfying the Markov Boundary Condition for G, then we say that G is a **perfect representation** of P. We will also say that P is **faithful** to G. We will henceforth assume that each probability distribution generated by a causal structure is perfectly represented by the causal graph G of that structure; justification for this assumption is provided in Spirtes[1990].

Pearl[1988] shows how to determine whether an atomic conditional independence statement is implied by the Markov boundary conditions for a graph G, using a graph-theoretic concept named d-separability.

An **undirected path from v_1 to v_n** in an acyclic graph G = <V,E> is an ordered n-tuple of vertices $\langle v_1, v_2, \ldots v_{n-1}, v_n \rangle$ such that each vertex occurs only once, and for each pair of vertices v_k and v_{k+1}, either the edge $\langle v_k, v_{k+1} \rangle$ is in E, or the edge $\langle v_{k+1}, v_k \rangle$ is in E. If, in an undirected path

$U = \langle v_1, v_2, \ldots v_{n-1}, v_n \rangle$ there is a vertex v_k where the edges $\langle v_{k-1}, v_k \rangle$ is in E, and $\langle v_{k+1}, v_k \rangle$ is in E then **v_k contains a direction-reversal in U.**

A **set of vertices X is d-separated from a set of vertices Y by a set of vertices Z** in a graph $G = \langle V, E \rangle$ iff there is no undirected path U between a variable in X and a variable in Y such that

a. for every vertex v_k on U that contains a direction-reversal in U, there is a directed path from v_k to some variable in Z, and

b. for every vertex v_k on U that does not contain a direction-reversal in U, v_k is not in Z.

Theorem(Pearl 1988): If G is a Bayes network of P, and X and Y are d-separated by Z in G, then X and Y are independent conditional on Z in P.

It follows that if P is faithful to G, then X and Y are independent conditional on Z in P if and only if X and Y are d-separated by Z in G.

4. Inference of Causal Structure

Let us call a set of variables V **causally sufficient** if every common cause of any pair of variables in V is also in V. In Fig. 1 the set V = {a, b, c, d, e, f} is causally sufficient, but the set V ' = {a, c, d, e, f} is not because b, a common cause of d and e, is not in V'.

Given a causally sufficient set of variables, and assuming that the graph of a causal structure is a perfect representation of any distribution generated by the causal structure, the following algorithm correctly constructs a set of models that includes the true causal structure.

PC Algorithm:

Let $A_C ab$ denote the set of vertices adjacent to a or to b in graph C, except for a and b themselves. Let $U_C ab$ denote the set of vertices in graph C on (acyclic) undirected paths between a and b, except for a and b themselves. (Since the algorithm is continually updating C, $A_C ab$ and $U_C ab$ are constantly changing as the algorithm progresses.)

A.) Form the complete undirected graph C on the vertex set V.

B.)

n = 0.

repeat

For each pair of variables a, b adjacent in C, if $A_C ab \cap U_C ab$ has cardinality greater than or equal to n and a, b are independent conditional on any subset S_{ab} of $A_C ab \cap U_C ab$ of cardinality n, delete a-b from C, and record S_{ab}.

n = n + 1.

until for each pair of adjacent vertices a, b, $A_C ab \cap U_C ab$ is of cardinality less than n.

C.) Let F be the graph resulting from step B. For each triple of vertices a, b, c such that the pair a, b and the pair b,c are each adjacent in F but the pair a, c are not adjacent in F, orient a - b - c as a -> b <- c if and only if b is not in S_{ac}. Output all graphs consistent with these orientations.

(Step C of the algorithm is an improvement upon our original algorithm suggested in Pearl and Verma [1990]. An algorithm that is similar in spirit but constructs undirected graphs has been independently suggested by Fung and Crawford[1990].)

The complexity of the algorithm for a graph G is bounded by $\max(|A_G ab|)$ over all pairs of vertices a,b, which is never more than the sum of the two largest degrees in G. Generally stage B of the algorithm continues testing for some steps after the correct undirected graph has been identified. The number of steps required before the true graph is found (but not necessarily until the algorithm halts) depends on the maximal number of treks[1] between a pair of variables, say a, b, that share no vertices adjacent to a or b. If these maximal numbers are held constant as the number of vertices increases, so that k, the maximal order of the conditional independence relations that need be tested, does not change, then the worst case computational demands of the algorithm increase as

$$\frac{n!}{2!(n-2)!} 2^k$$

which is bounded by n^2. It should be possible to recover sparse graphs with as many as several hundred variables. Of course the computational requirements increase exponentially with k.

[1] A trek is a pair of directed paths from some vertex z to a, b respectively, intersecting only at z, or a directed path from a to b or a directed path from b to a.

In many cases it is more efficient to perform conditional independence tests on all subsets of $A_G ab$ rather than to compute $U_G ab$. We have not yet theoretically determined the trade-off.

The structure of the algorithm and the fact that it continues to test even after having found the correct graph suggest a natural heuristic for very large variable sets whose causal connections are expected to be sparse, namely to set a fixed bound on the order of conditional independence relations that will be considered.

Theorem: If the causal graph G that generated a distribution P is causally sufficient and a perfect representation of P, then given a list of the conditional independence relations true of P as data, the PC algorithm constructs a set of graphs that includes the true graph.

The PC algorithm has two major advantages over other algorithms that have been suggested for discovering causal structures.

First, it takes advantage of the sparseness of the graph to reduce the number of conditional independence relations that need to be tested.

Second, it can be reliably applied to large numbers of variables even if the sample size is only moderately large. If A and B are independent conditional on C, let us call the cardinality of C the order of the conditional independence. For discrete variables, reliable tests of high order conditional independence relations requires huge sample sizes. Because the PC algorithm takes advantage of the sparseness of the graph to reduce the order of the conditional independence relations that need to be tested, it can be applied to large sets of variables with only moderate sample sizes.

For example, we have applied the PC algorithm to simulated data generated from the following causal network taken from Beinlich[1989].

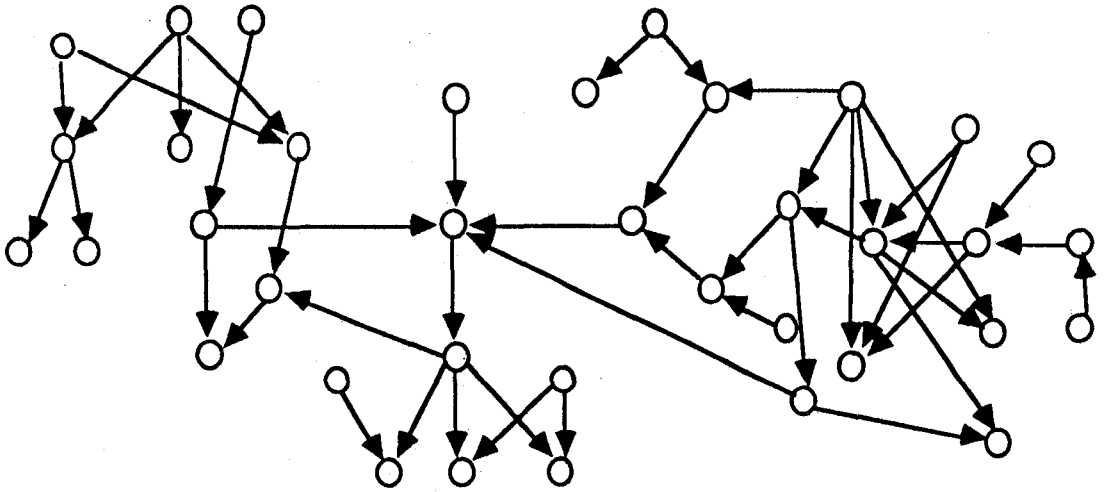


Fig. 2: Alarm Network

Results were scored separately for errors concerning the existence and the directions of edges, and for correct choice of regressors. An edge existence error of commission (Co) occurs when any pair of variables are adjacent in the output but not in the pattern of graph (b) in figure 8. An edge direction error of commission occurs when any arrowhead not in the pattern of (b) occurs in the output in an edge occuring in the pattern of (b). Errors of omission (Om) are defined analogously in each case. The results are tabulated as the average over the trial distributions of the ratio of the number of actual errors to the number of possible errors of each kind. The results were as follows:

Variable	#trials	n	%Edge Existence Errors		%Edge Direction Errors	
			Co	Om	Co	Om
Linear	3	10,000	2.7	4.3	23.7	3.7
Discrete	1	10,000	0.3	6.5	14.0	8.3

5. Partial Correlations and D-Separability

The input to the PC algorithms requires determining when a given conditional independence relation holds in the distribution P. However, in the case of linear causal theories, rather than using facts about conditional independencies in P as input, we use facts about vanishing partial correlations. The following theorems justifies using the results of statistical tests of vanishing partial correlations as input to the PC algorithm.

For a linear causal theory with graph G, let us say that a partial correlation $\rho_{xz.Y}$ is **strongly implied** to vanish if and only if it vanishes for every linear distribution generated by G. We assume that any partial correlation that vanishes in the population is strongly implied to vanish because of the following theorem.

Theorem: Let M be a linear model with n free linear coefficients $a_1,..., a_n$. and k variances $v_1,..., v_k$. Let M(U) be the model obtained by specifying values $U = \langle u_1,...,u_n, u_{n+1},...,u_{n+k}\rangle$ for $a_1,..., a_n$ and $v_1,...v_k$. Let $\mathbf{P}$ be the set of probability measures P on the space R^{n+k} of values of the parameters of model M such that for every subset S of R^{n+k} having Lebesque measure zero, P(S) = 0. Let Q be the set of vectors of coefficient and variance values such that for all U in Q every multinormal probability distribution consistent with M(U) has at least one statistical independence relation not represented in the directed acyclic graph of M according to d-separability. Then for P $\in \mathbf{P}$, P(Q) = 0.

We have also proved the following:

Theorem: In a linear causal system with graph G and distribution P, if x and z are distinct variables, and Y is a set of variables not including x and z, then Y d-separates x and z if and only if $\rho_{xz.Y}$ is strongly implied to vanish.

6. Mixed Causal Structures

Consider a population that is a mixture of structures <g,P1> and <g,P2> where P1 and P2 are distinct and, we will suppose, both faithful to graph g. Let the proportions in the mixture by n:m. This sort of case appears to be the simplest and easiest sort of mixing, for we know in this case that the two distributions have the very same conditional independence relations. The unfortunate fact, however, is that even in this case the mixed population does not generally have those same conditional independence relations, and indeed unless special constraints are satisfied by P1 and P2, the mixed distribution will have no non-trivial conditional independence relations at all.

In 1903 G. Udny Yule[2] concluded his fundamental paper on the theory of association of attributes in statistics with a section "On the fallacies that may be caused by the mixing of distinct records", (where |AB| is a measure of associate between A and B that vanishes when A and B are independent):

[2]G. U. Yule, "Notes on the Theory of Association of Attributes in Statistics" *Biometrika*, 121-133.

It follows from the preceding work that we cannot infer independence of a pair of attributes within a sub-universe from the fact of independence within the universe at large...The theorem is of considerable practical importance from its inverse application; i.e. even if |AB| have a sensible positive or negative value we cannot be sure that nevertheless |AB | C | and |AB | ~C | are not both zero. Some given attribute might, for instance, be inherited neither in the male line nor the female line; yet a mixed record might exhibit a considerable apparent inheritance.

The fictitious association caused by mixing records finds its counterpart in the spurious correlation to which the same process may give rise in the case of continuous variables, a case to which attention was drawn and which was fully discussed by Professor Pearson in a recent memoir. If two separate records, for each of which the correlation is zero, be pooled together, a spurious correlation will necessarily be created unless the mean of one of the variables, at least, be the same in the two cases.

Let $P(XYZ) = nP_1(XYZ) + mP_2(XYZ)$, with $n + m = 1$. Elementary algebra shows that $P(XY/Z) = P(X/Z)P(Y/Z)$ if and only if

$$n^2P_1(XYZ)P_1(Z) + nmP_2(XYZ)P_1(Z) + mnP_1(XYZ)P_2(Z) + m^2P_2(XYZ)P_2(Z) =$$

$$n^2P_1(XZ)P_1(YZ) + nmP_1(XZ)P_2(YZ) + mnP_2(XZ)P_1(YZ) + m^2P_2(XZ)P_2(YZ).$$

If in both distributions, X,Y are independent conditional on Z, that is $P_1(XY/Z) = P_1(X/Z)P_1(Y/Z)$ and $P_2(XY/Z) = P_2(X/Z)P_2(Y/Z)$, then the equation above reduces to

$$P_2(XYZ)P_1(Z) + P_1(XYZ)P_2(Z) = P_1(XZ)P_2(YZ) + P_2(XZ)P_1(YZ)$$

which is not a function of the proportions n, m. This equation can be put in the slightly more perspicuous form:

$$P_2(XY/Z) + P_1(XY/Z) = P_1(X/Z)P_2(Y/Z) + P_2(X/Z)P_1(Y/Z).$$

or, since we are assuming that X and Y are conditionally independent on Z in both P1 and P2,

$$P_2(X/Z)P_2(Y/Z)+P_1(X/Z)P_1(Y/Z) = P_1(X/Z)P_2(Y/Z)+ P_2(X/Z)P_1(Y/Z)$$

The rather surprising conclusion is that when we mix probability distributions we should expect to find all possible conditional *dependence* relations. Hence in mixed populations, conditional independence and dependence will not be a reliable guide to causal structure. Applying the PC algorithm to such data will, for example, produce a complete undirected graph. This has a practical if informal moral for the significance we ought to give to inferences from non-experimental data. When from properly collected data sets with large sample sizes we find that the resulting undirected graph is not complete, we ought to be a little impressed. Either some constraints have been satisfied by chance, or over some variables almost all units in the sample have the same causal structure, and that structure does not include the missing connection.[3]

In the case of linear structures with faithful probability distributions, independence is marked by vanishing correlations and conditional independence by vanishing partial correlations. When populations with two different distributions each associated with a linear structure are mixed, vanishing correlations in the mixed distribution will not mark independence in the mixed distribution, and vanishing partial correlations in the mixed distribution will not mark conditional independence in the mixed distribution. It is easy to verify that for any mixture of two distributions--based on linear structures or not--the covariance of two variables vanishes in the mixture if and only if

$$k_1COV_1(XY) + k_2COV_2(XY) =$$

$$k_1k_2[\mu_1X\,\mu_2Y + \mu_1Y\,\mu_2X] + k_1(k_1 -1)\mu_1X\mu_1Y + k_2(k_2 -1)\mu_2X\mu_2Y]$$

where the proportion of population 1 to population 2 is n: m and k1 = $n/(n+m)$, k2 = $m/(n+m)$.

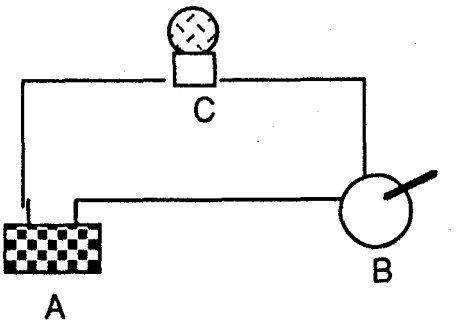
Some of the uncertainty occasioned by mixed populations disappears when we can impose experimental controls, and this is one of the principal advantages of experimental procedures.

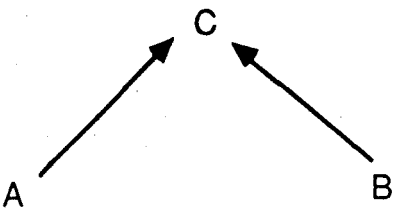
---

[3]An interesting question is therefore whether there is any means to infer causal dependence in mixed populations or samples. Suppose we have prior knowledge that excludes the complete undirected graph; it may even forbid particular edges. If we then obtain a probability distribution in which no non-trivial conditional independence relations hold, can we infer anything about the class of mixtures (consistent with the prior knowledge) from which the population distribution may have come? We have no idea.

A population of systems sharing the same causal structure but considered with regard to a set V of variables that fails to include some common cause T of variables in V might be considered to have a mixed distribution. If, for simplicity, T is binary, the population can be viewed as a mixture of a subpopulation in which T = 1 and a subpopulation in which T = 0. Assuming for simplicity that there is no other causal connection between X and Y besides a trek with T as its source, in the subpopulation with T = 0 the variables X and Y will be independent and in the subpopulation with T = 1 X and Y will be independent, but in the mixed population X and Y will of course be dependent.

When we consider discrete data, the phenomena of mixtures show a fundamental limitation in our means of representing causal relations. Consider a simple switch. Suppose battery A has two states: charged and uncharged. A charge in battery A will cause bulb C to light up provided the switch B is on, but not otherwise.





If A and B are independent random variables, then A and C are dependent conditional on B and on the empty set, and B and C are dependent conditional on A and the empty set, and A and B are dependent conditional on C. The causal structure therefore looks like the directed graph shown above.

There is nothing wrong with this conclusion except that it is not fully informative. The dependence of A and C arises entirely through the condition B = 1. When B = 0, A and C are independent. The

graph does not tell us that when B = 0 manipulating A will have no effect on C. Knowledge of the causal graph without this further information could lead to very mistaken expectations. Consider, for example, cases in which "switch" variables analogous to B have off values in the vast majority of the population. Then manipulating causes such as A will in most cases have no effect. Since in discrete data the conditional independence facts, if known, identify the switch variables, a better representation would identify certain parents of a variable as switches. But because in variables that take several discrete values, A may be a switch for B and B may be a switch for A, and several distinct values of B may be "on" values for A and conversely, a general representation of this sort would often not be very easy to grasp.[4]

7. Random Coefficient Linear Structures

In section 2, we described linear causal structures in which each unit in the population has the same linear coefficients (i.e. they were constant random variables). Let us now call this a **constant coefficient linear causal structure**. In a **random coefficient linear causal structure**, the coefficients are non-constant random variables such that any set of coefficients is independent of any other disjoint set of coefficients or non-coefficient exogenous random variables. Note that this independence assumption is also true of constant coefficient linear causal structures.

Theorem: For any two variables x and y in a random coefficient linear causal structure RC, the covariance of x and y is equal to the covariance of the corresponding variables x' and y' in a constant coefficient linear causal structure CC with the same graph, and in which the expected value of each linear coefficient in RC is equal to the constant value of the corresponding linear coefficient in CC.

Proof. The covariance of x and y is equal to $E(x\,y) - E(x)\,E(y)$. The formulas occurring in the following proof are correct for both models RC and CC.

If there is an edge from non-coefficient random variable *a* to *b* in the graph, then there is a non-zero coefficient of *a* in the equation for *b*. Label the edge from *a* to *b* by this non-zero coefficient. Label a directed path p by the product of the labels of the edges in the path. Let Ex be the set of exogenous variables, and P_{ab} be the set of paths from *a* to *b*.

The values of random variables x, y, and xy are respectively:

---

[4]A better practical arrangement might be a query system that, besides inferring the causal graph or graphs, responds to the user's questions about the effects of the manipulation of variables.

$$x = \sum_{e \in Ex} \sum_{p \in P_{ex}} L(p)e$$

$$y = \sum_{f \in Ex} \sum_{q \in P_{fy}} L(q)f$$

$$xy = \sum_{e \in Ex} \sum_{p \in P_{ex}} \sum_{f \in Ex} \sum_{q \in P_{fy}} L(p)L(q)ef$$

The expected values of each of these variables is given below:

$$E(x) = E\left(\sum_{e \in Ex} \sum_{p \in P_{ex}} L(p)e\right) =$$

$$\sum_{e \in Ex} \sum_{p \in P_{ex}} E(L(p)e)$$

$$E(y) = E\left(\sum_{f \in Ex} \sum_{q \in P_{fy}} L(q)f\right) =$$

$$\sum_{f \in Ex} \sum_{q \in P_{fy}} E(L(q)f)$$

$$E(xy) = E\left(\sum_{e \in Ex} \sum_{p \in P_{ex}} \sum_{f \in Ex} \sum_{q \in P_{fy}} L(p)L(q)ef\right) =$$

$$\sum_{e \in Ex} \sum_{p \in P_{ex}} \sum_{f \in Ex} \sum_{q \in P_{fy}} E(L(p)L(q)ef)$$

Since the coefficients are independent of each other and the non-coefficient random variables,

$$E(L(p)\, L(q)\, e\, f) = E(L(p))\, E(L(q))\, E(e\, f).$$

The label of a path is equal to the product of the labels of the edges

$$L(p) = \prod_{edge \in p} L(edge)$$

Substituting into the formula for $E(x\, y)$, we obtain

$$E(xy) = \sum_{e \in Ex} \sum_{p \in P_{ex}} \sum_{f \in Ex} \sum_{q \in P_{fy}} E\left(\prod_{edge \in p} L(edge)\right) E\left(\prod_{edge \in q} L(edge)\right) E(ef)$$

By the independence of the linear coefficients,

$$E(\prod_{edge \in p} L(edge)) = \prod_{edge \in p} E(L(edge))$$

It follows that

$$E(xy) = \sum_{e \in Ex} \sum_{p \in P_{ex}} \sum_{f \in Ex} \sum_{q \in P_{fy}} \prod_{edge \in p} E(L(edge)) \prod_{edge \in q} E(L(edge)) E(ef)$$

Similarly,

$$E(L(p)\ e) = E(L(p))E(e)$$

and

$$E(L(q)\ f) = E(L(q))E(f).$$

Substituting these into the formula for $E(x)\ E(y)$ we obtain

$$E(x)\ E(y) = \left(\sum_{e \in Ex} \sum_{p \in P_{ex}} E(L(p))\ E(e) \right) \left(\sum_{f \in Ex} \sum_{q \in P_{fy}} E(L(q))\ E(f) \right)$$

Again, from the independence of the linear coefficients if follows that

$$E(x)\ E(y) = \left(\sum_{e \in Ex} \sum_{p \in P_{ex}} \prod_{edge \in p} E(L(edge))\ E(e) \right) \left(\sum_{f \in Ex} \sum_{q \in P_{fy}} \prod_{edge \in q} E(L(edge))\ E(f) \right)$$

In CC, since *L(edge)* is a constant, *E(L(edge)) = L(edge).* In RC, by hypothesis, *E(L(edge)) = L(edge)* in CC. Hence the expression *E(x y) - E(x) E(y)* is the same in both RC and CC. Q.E.D.

Note that this proof depends upon the independence of the linear coefficients from each other. This is true of both constant coefficient and random coefficient linear causal structures, but not true in general of linear causal structures in which large sub-populations share the same linear coefficients.

The covariances completely determine the values of the partial correlations. Hence, we can apply the theorems of section 5 to justify using facts about vanishing partial correlation as input to the PC algorithm for random coefficient linear causal structures.

GAUSSIAN WINDOWS:
A TOOL FOR EXPLORING MULTIVARIATE DATA

Louis A. Jaeckel

Research Institute for Advanced Computer Science
Mail Stop Ellis Street
NASA Ames Research Center
Moffett Field, CA 94035-1000

RIACS Technical Report 90.41
September 1990

INTRODUCTION

Given a large set of quantitative multivariate data, say, N data points in a p-dimensional space:

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{ip}), \qquad i = 1 \text{ to } N,$$

we want to explore the structure of the data. That is, we want to find the shape of the underlying density function. We assume that the density function is more or less smooth.

To explore the data, we need a way to look at the local structure of the data in a limited region. So we will examine the data in a given region by viewing the data through a GAUSSIAN WINDOW, whose location and shape are chosen by the user. By doing this we will be able to find and describe simple structural features in the data in any number of dimensions. Some examples are given on the next page.

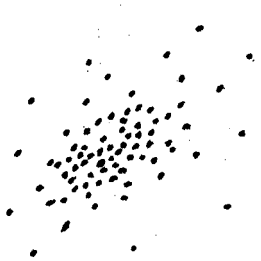
We will describe the local structure of the data by a method similar to the method of principal components.

By taking many local views of the data, we can build up an idea of the structure of the data set. That is, the method is INTERACTIVE. With practice, we can apply our geometrical intuition to the features we find in the data, in any number of dimensions.

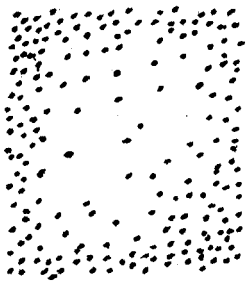
Since the computations are relatively simple, the method can be implemented on a small computer.

EXAMPLES OF STRUCTURAL FEATURES IN TWO DIMENSIONS

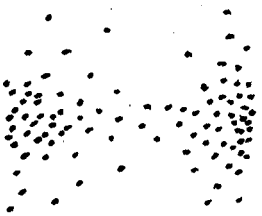
Some examples of the kinds of structures that we can find and describe are the following:



A peak, or cluster

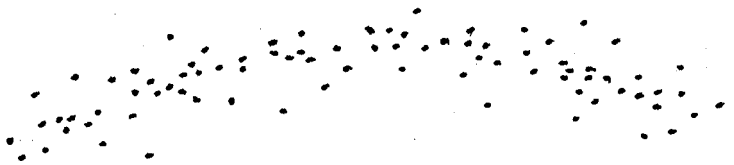


A valley



A saddle point

We can also find extended structures such as a "ridge" or "bar":



Only a part of an extended structure would be visible in a single window. If part of such a structure appears in a window, we can tell that we are looking at a structure that extends beyond the window. We can then follow along it and map out its extent and shape.

We can imagine analogous structures in higher dimensions. For example, a "ridge" in p dimensions is an essentially one-dimensional structure, consisting of data points concentrated near a "center line" but scattered about it in all directions.

Similarly, we might find an essentially k-dimensional structure in a p-dimensional space, for any $k < p$.

THE GAUSSIAN WINDOW

To focus on a limited region in the space, we use a window.

A GAUSSIAN WINDOW is defined by choosing a center point α and a non-negative definite symmetric matrix V to describe its size and shape. Let

$$w(x) = e^{-\frac{1}{2}(x - \alpha)'V(x - \alpha)} ,$$

where x is a p-vector and "prime" means "transpose".

Each data point x_i is given the weight $w_i = w(x_i)$. Note that $w(x) \leq 1$ for all x, and that $w(x)$ decreases as x moves away from α. Thus we have defined a window with "fuzzy" boundaries.

We then compute a vector called the WEIGHTED SAMPLE MEAN,

$$\bar{x}_w = \frac{1}{\Sigma w_i} \Sigma w_i x_i ,$$

and a matrix called the WEIGHTED SAMPLE COVARIANCE MATRIX,

$$S_w = \frac{1}{\Sigma w_i} \Sigma w_i (x_i - \bar{x}_w)(x_i - \bar{x}_w)' .$$

We also compute $\frac{1}{N} \Sigma w_i$.

These quantities are the simplest things to compute, especially in a high-dimensional space. They describe the overall shape of the weighted data in the "window region" (the region vaguely defined as the region where $w(x)$ is "not small"). The estimated shape of the density function in the window region will be based on these quantities. Note that they are overall statistics; any "fine structure" in the region is smeared out.

To look for finer details, we would use smaller windows.

EXAMPLE: A CLUSTER

Suppose that in the region of a window, the density function has approximately a multivariate Gaussian shape:

$$f(x) = c \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x - \mu)'\Sigma^{-1}(x - \mu)} ,$$

where μ, Σ, and c are all unknown parameters. That is, we have a single peak (or cluster of data points) in the window region.

The vector μ is the center point of this part of the density.

The symmetric matrix Σ is its covariance matrix.

The constant c represents the "probability mass" of this part of the entire probability distribution.

Let $B = \Sigma^{-1}$ and let $a = c \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}$. Then

$$f(x) = a\, e^{-\frac{1}{2}(x - \mu)'B(x - \mu)} .$$

The WINDOWED DENSITY FUNCTION, the effective density function of the data as viewed through the window, is $w(x)f(x)$. That is, if we assign weight $w_i = w(x_i)$ to each data point x_i, and if we do computations with the weighted x_i, the results will be as if we were working with a sample from $w(x)f(x)$.

Assume for simplicity that α, the window center, is 0.

Let $A = B + V$. Then, by doing some algebra, we find that the windowed density function $w(x)f(x)$ is

$$\left[a\, e^{-\frac{1}{2}\mu'BA^{-1}V\mu} \frac{(2\pi)^{p/2}}{|A|^{1/2}} \right] \frac{|A|^{1/2}}{(2\pi)^{p/2}} e^{-\frac{1}{2}(x - A^{-1}B\mu)'A(x - A^{-1}B\mu)} .$$

This is a multivariate Gaussian function with "windowed mean" $A^{-1}B\mu$ and "windowed covariance matrix" A^{-1}.

ESTIMATION OF PARAMETERS

It follows that the weighted sample mean $\bar{x}_w$ is an estimate of $A^{-1}B\mu$, and the weighted sample covariance matrix S_w is an estimate of A^{-1}. The expression in the square brackets (bottom of Page 5) is the integral of $w(x)f(x)$ over the entire space. We will estimate it by the average of the weights: $\frac{1}{N}\Sigma w_i$.

We now DEGAUSS the view of the data as seen through the Gaussian window. That is, we remove the effect of the weights on the shape of the data in the window region. Since S_w is an estimate of A^{-1}, we can estimate A by S_w^{-1}, and we have

$$S_w^{-1} = \hat{A} = \hat{B} + V \ .$$

So we can estimate B by

$$\hat{B} = S_w^{-1} - V \ .$$

We can then estimate Σ by

$$\hat{\Sigma} = \hat{B}^{-1} = (S_w^{-1} - V)^{-1} \ ,$$

assuming that $S_w^{-1} - V$ is positive definite.

Since $\bar{x}_w$ is an estimate of $A^{-1}B\mu$, we can estimate μ by

$$\hat{\mu} = \hat{B}^{-1}\ \hat{A}\ \bar{x}_w = (S_w^{-1} - V)^{-1}\ S_w^{-1}\ \bar{x}_w \ .$$

Since $\frac{1}{N}\Sigma w_i$ is an estimate of the expression in the square brackets, we can also estimate the constants a and c.

These estimated parameters give us an estimate of the shape of the density function in the window region. Note that the computations are standard matrix operations.

RELATION TO PRINCIPAL COMPONENTS ANALYSIS

If we find a cluster in a window, we can describe its shape using the method of principal components. To do this we find the eigenvalues and corresponding eigenvectors of $\hat{\Sigma}$.

The estimated shape of the cluster is a p-dimensional ellipsoidal shape centered at $\hat{\mu}$. The principal axes of the ellipsoid are parallel to the eigenvectors. The estimated density function can be expressed as a product of p univariate Gaussian (normal) densities, each lying along a principal axis. The standard deviation of each of these densities is the square root of the corresponding eigenvalue (all of which are positive in this case). Thus we have a way of thinking about the shape of the cluster in any number of dimensions.

Note that we could do this analysis based on the matrix $\hat{B}$, which is the inverse of $\hat{\Sigma}$. These two matrices have the same eigenvectors, and the eigenvalues of $\hat{B}$ are the reciprocals of those of $\hat{\Sigma}$. When we deal with structures other than clusters, we will analyze their shape by looking at the eigenvalues and eigenvectors of $\hat{B}$.

For example, if the shape of the density function in the window region is a valley or a saddle point, then all or some of the eigenvalues of $\hat{B}$ will be negative. A negative eigenvalue indicates that, in the window region, the density function is concave upward along the direction of the corresponding eigenvector.

Since the eigenvalues of $\hat{B}$ are the reciprocals of those of $\hat{\Sigma}$, an eigenvalue of $\hat{B}$ near 0 indicates a structure extending beyond the window region, and a large positive eigenvalue indicates that the data points are tightly concentrated along the corresponding direction.

MORE GENERAL STRUCTURAL FEATURES

An example is a "ridge", or a "bar", which is an essentially one-dimensional concentration of points. We will assume that the density function in the window region can be approximated by

$$f(x) = h\, e^{-\frac{1}{2} x'Bx + r'x} .$$

This is a general expression which includes the cluster example above, and also the other examples on Page 3. The exponent is a general polynomial of degree two in the coordinates of the vector x.

The constant h is the density at the window center (assumed to be at 0). The symmetric matrix B may or may not be positive definite, and it may or may not be non-singular. If B is singular, there is no center point μ for the function.

As before, the windowed density function $w(x)f(x)$ is a multivariate Gaussian function.

We compute $\bar{x}_w$, S_w, and $\frac{1}{N}\Sigma w_i$ as before, and we estimate the parameters B, r, and h based on these quantities.

In the general case, $\hat{B}$ might be nearly singular, so we won't invert it. Instead, we will work directly with $\hat{B}$.

We use a method similar to principal components analysis: We find the eigenvalues and eigenvectors of $\hat{B}$, and we use these quantities to describe the shape of the estimated density function in the window region. As in principal components analysis, we can express the estimated density function as a product of p functions of one variable each. The interpretation of the eigenvalues is the same as on the previous page.

EXAMPLE: A RIDGE OR BAR

We can see part of such an extended structure in a window, and we can tell by the existence of some eigenvalues near 0 that it extends beyond the window. In the case of a ridge, $\hat{B}$ will have one eigenvalue very near 0, and the corresponding eigenvector will be parallel to the center line, or crest, of the ridge. The small eigenvalue indicates that the data in the window region appear to have an essentially "infinite" variance in the corresponding direction.

Since a structure like this does not have a center point, as a cluster does, we will not to try to estimate a center point here.

Instead, we will estimate the location of the center line of the ridge, and we will estimate the shape of the cross-section of the ridge. Note that in a p-dimensional space, a ridge would have a (p-1)-dimensional cross-section orthogonal to the center line.

If we find a structure like this, we can then move the window center to the nearest point on the center line and try another window. Then we can follow along the ridge by moving the window center along the estimated center line. By continuing in this way we can map out the extent and shape of the ridge.

An essentially k-dimensional structure, or concentration of data points, in a p-dimensional space can be treated in a similar way.

PROPERTIES OF THE METHOD

Since it is interactive, it is flexible and open-ended.

It can be used (in principle) in any number of dimensions.

Few assumptions are made about the data.

We can search for structural features by trying many different windows, and we can describe the features we find. Then we can put together what we have found into an overall description of the data.

The method can be used in conjunction with other methods, such as graphical methods that involve projecting the data onto a space of lower dimension, and automatic methods such as clustering algorithms. Note that with this method we can find structural features other than clusters.

Since the computations are relatively simple, the method can easily be implemented on a small computer. Any standard algorithms for inverting a matrix and for finding the eigenvalues and eigenvectors of a symmetric matrix can be used. (I wrote a simple program in BASIC on an IBM PC to test the method, and I have done some experiments with a number of artificial data sets.)

Most importantly, we can apply our geometric intuition to the features we find in the data, so that we can think about and describe the structure of a set of data in any number of dimensions.

Learning Classification Trees

Wray Buntine
wray@ptolemy.arc.nasa.gov
RIACS & NASA Ames Research Center
Mail Stop 244-17, Moffet Field, CA 94035, USA

Abstract

Algorithms for learning classification trees have had successes in artificial intelligence and statistics over many years. These notes outlines how a tree learning algorithm can be derived from Bayesian decision theory. This introduces Bayesian techniques for splitting, smoothing, and tree averaging. The splitting rule turns out to be similar to Quinlan's information gain splitting rule, while smoothing and averaging replace pruning. Comparative experiments with reimplementations of a minimum encoding approach, Quinlan's C4 and Breiman *et al.*'s CART show the full Bayesian algorithm is consistently as good, or more accurate than these other approaches though at a computational price.

These notes present material from NASA Ames Artificial Intelligence Research Branch Technical Report FIA-90-12-19-01, *Learning Classification Trees*, by Wray Buntine.

Bayesian Trees: A Theoretical and Empirical Comparison of Learning Theories

Wray Buntine

RIACS
NASA Ames Research Center

• acknowledgements: Ross Quinlan, Robin Hanson, Peter Cheeseman, ...
• this is a rationalisation and extension of work in my PhD thesis
• overview papers are available

Outline

- what is a tree?
- motivation
- background
- theory
- implementation
- experimental results
- theory comparison
- extensions

aims of this talk: to pass on intuitions gained without getting bogged down in the math

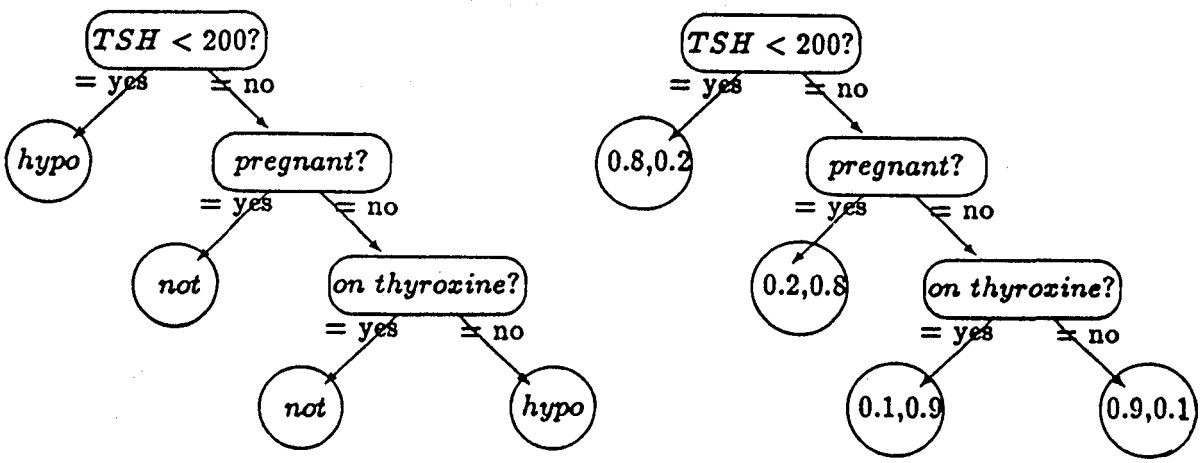


Figure 1: A decision tree and a class probability tree from the thyroid application

NB. class prob. tree is a class conditional probability distribution

The Learning Problem

from training sample (i.i.d. examples), e.g.

example (*TSH*	*pregnant*	*age*	*on-thyroxine*)	*class*
175	yes	35	no	hypo
210	no	46	yes	not
180	yes	51	yes	hypo
230	yes	42	no	not
240	no	47	no	hypo

I construct a class probability tree that should predict future classifications well

or II construct a classifier that should predict future classifications well

(this ignores comprehensibility & efficiency issues)

Problems with Empirical Learning Research Through the Ages

(statistics, AI, neural nets, numerous application areas, ...)

Empirical learning: learning classifiers, models or other forms of knowledge by analysing data such as historical records

Problems with research:

- "feel-good" algorithms
- dubious theories (unclear assumptions, etc.)
- soft testing, use of straw-men, etc.
- poor literature review

Research Goals

(from hindsight)

- to understand how to design learning algorithms for different learning problems
  - to understand and compare different learning theories (e.g. MDL, Bayesian, uniform convergence)
  - to understand when and how approximations should be made

Why learn trees?

(from hindsight)

Tree algorithms are old hat, fairly mature, why bother developing more? Especially when better gains are to be made by extending the model space?

- hard enough learning problem to be interesting (e.g. tackles problem of overfitting)
- great case study to compare theories and develop algorithms (due to existing competition)
- good starting point for many related problems e.g. Bayesian networks, variable n-gram models, regression, etc.

Background

CART: "classification and regression trees", by Brieman, Friedman, Olshen and Stone, 1980-1984

ID3 & C4: developed by Quinlan, 1979-1988, with numerous commercial spin-offs

MML & MDL: minimum encoding methods by Quinlan and Rivest, Rissanen, and Wallace, 1987-89

other information theory, AI, statistics ...

Notation

Pr(X | Y) = subjective belief that X is true given you know just Y is true

Pr satisfies same properties as a frequency, and is measured in units of probability

Pr is best interpreted as a relative quantity

i.e. "Pr(tree) > Pr($\overline{\text{tree}}$)/9 ?" is OK

"Pr(tree) > 0.1 ?" is not

Bayesian Theory Outline

class probability tree

= conditional probability distribution for class

= T + θ

= tree structure + class probabilities (at leaves)

= discrete comp. + continuous comp.

assume **sample** is independently and identically distributed set of completely-specified classified examples

Bayesian solution in a nutshell: wish to determine

Pr(class | new-example, sample)

$$= \sum_T \int_\theta \Pr(\text{class} \mid \text{new-example}, T, \theta)\ \Pr(T, \theta \mid \text{sample})$$

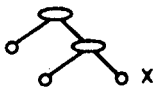
$$= \sum_T \Pr(\text{class} \mid \text{new-example}, T, \text{sample})\ \Pr(T \mid \text{sample})$$

then combine this with appropriate loss function

Bayesian Tree Learning

NB. Pr(class | new-example, sample)

$$= \sum_T \Pr(\text{class} \mid \text{new-example}, T, \text{sample}) \Pr(T \mid \text{sample})$$

tree structure + tests	posterior probability	class probability for new example
	0.001	0.54
	0.025	0.87
	0.001	0.67
etc.	...	...
	1.000	0.80 = weighted average class probability

Intuition:

- take weighted average of "representative set" of trees
- weights = posterior probabilities, determined using weights of evidence

Theory Comparison from Bayesian Viewpoint

Full Bayesian

Posterior class probability for new example =

$$\sum_T \Pr(\text{class} \mid \text{new-example}, T, \text{sample}) \Pr(T \mid \text{sample})$$

Minimum Encoding (e.g. MDL)

1. Use a particular "coding" form for Pr(T) (i.e. prefer simpler structures)
2. Choose tree structure T maximising Pr(T | sample)

Resampling (e.g. CART)

1. Construct a sequence of trees (but not too many).
2. Manufacture "many psuedo-independent" training/test pairs and use these to estimate risk for each tree, then pick the best

Uniform Convergence (basis of most computational learning theory)

1. Ensure sample is large enough so single tree (and its variants) dominate the posterior sum.
2. Choose tree structure T minimising risk(sample|T)

   NB. If hypothesis space is correct, this is identical to maximising Pr(T | sample)

Modelling

Model

T = structure of tree including shape and tests at interior nodes

$\theta_{c|l}$ = proportion of class c at leaf l

Priors

$Pr(T,\theta)$ = prior on T and θ = $Pr(T)\ Pr(\theta|T)$

where

$$Pr(\theta|T) = \prod_{l \text{ in leaves}} \frac{\prod_{c \text{ in classes}} \theta_{c|l}^{\alpha}}{Beta(\alpha,\alpha,...,\alpha)}$$

a Dirichlet distribution

$Pr(T)$ = constant

or = belief slightly favours smaller trees

or = belief strongly favours smaller trees (e.g. as determined using "coding" of tree)

Bayesian Analysis

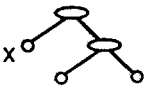
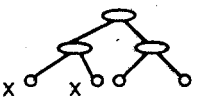
Posterior probability

let $n_{c|l}$ = count from sample of examples at leaf l in class c

$$Pr(T|sample) = Pr(T) \prod_{l \text{ in leaves}} \frac{Beta(n_{1|l}+\alpha,\ n_{2|l}+\alpha,...,\ n_{C|l}+\alpha)}{Beta(\alpha,\alpha,...,\alpha)}$$

Comparative heuristic

given tree T, if we replace a leaf node by a test with several leaves, to get tree T+

T =  → T+ = 

the log-odds, log (Pr(T+|sample)/Pr(T|sample)), is heuristic assessment of the quality of the replacement

Overview of Implemented Approximations

To be approximated:

Posterior sum = $\sum_T$ Pr(class | new-example,T,sample) Pr(T | sample)

Growing: posterior probabilities of sub-structures give heuristic measures of sub-structure quality in units of log-odds or probability; they can therefore be used for:

- significance testing of comparative quality of different sub-structures
- significance testing of whether to stop growing
- randomised growing of trees for monte-carlo approximations

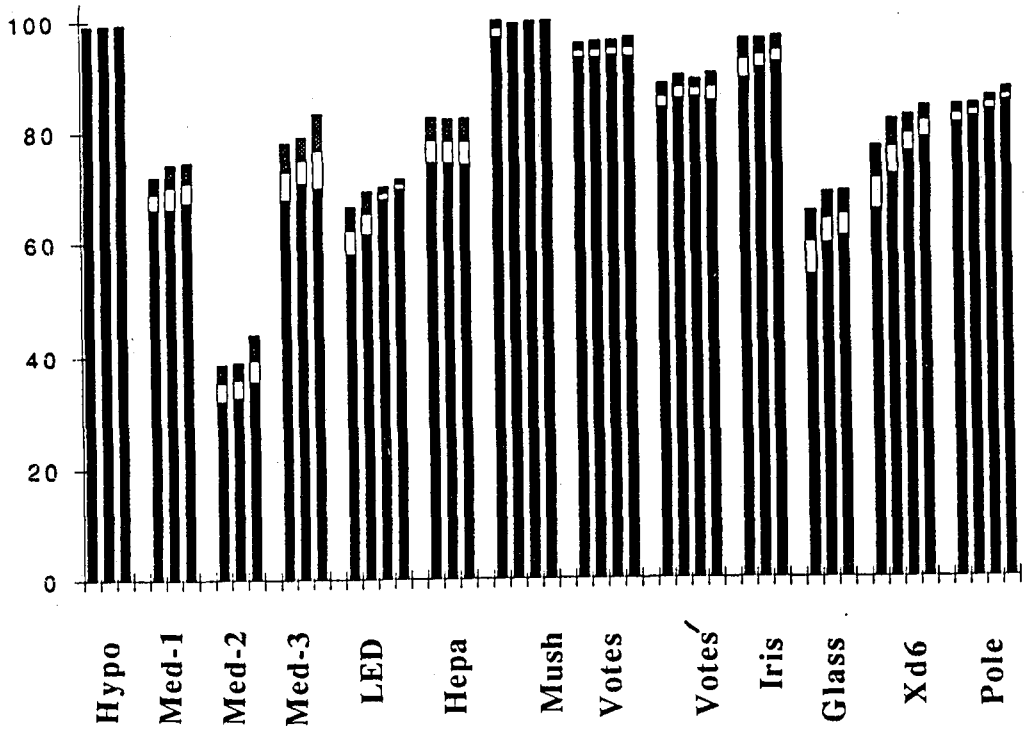
Smoothing: a subset of the posterior sum can sometimes be computed in closed form; this turns out to be analogous to smoothing techniques

Averaging: represent many structures in an AND-OR form to compactly represent the dominant terms in the posterior sum

Multiple Models: make a monte-carlo approximation (using importance sampling) of the posterior sum by generating structures randomly according to their posterior

Multiple Trees

Situation	Representative set
have lots of data + good tree classifier exists...	single dominant tree
less data exists.....	several different good trees
trees are a poor model and less data exists.....	lots of lousy trees
data is "pure noise" (so no good tree exists).....	even more lousy trees



Average (across all algorithms) accuracies for different sample sizes grouped by domain.

Experiments

Tree Methods

CART, C4 and MML: reimplementations that perform comparably with the originals

Mult.: consists of building 5 trees (each built by randomly selecting tests at nodes according to their posterior probability), and then averaging their predictions

Ave. (*n*-ply): Bayesian averaging and smoothing using *n*-ply lookahead (above algorithms all use 1-ply)

Notes

- chose data sets to get broad variety of problems, chose training sets to show cross section of learning curve
- took average of 20 random train/test pairs, tested significance using paired *t*-test

Results

- CART and C4 are roughly comparable in performance; MML is sometimes worse as it usually overprunes; all 3 approaches usually overprune with highly structured data
- Ave. with 1 or 2-ply is usually as good or significantly better; Mult. is competitive

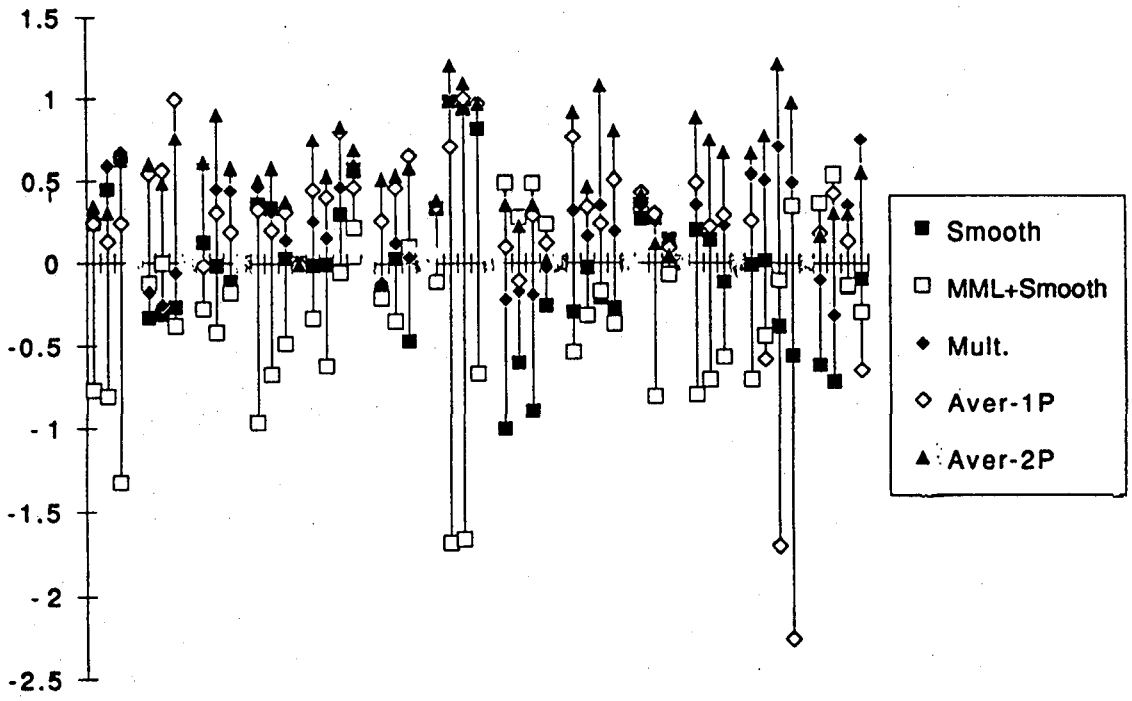


Figure 5: Normalised Accuracies for Bayesian Variations

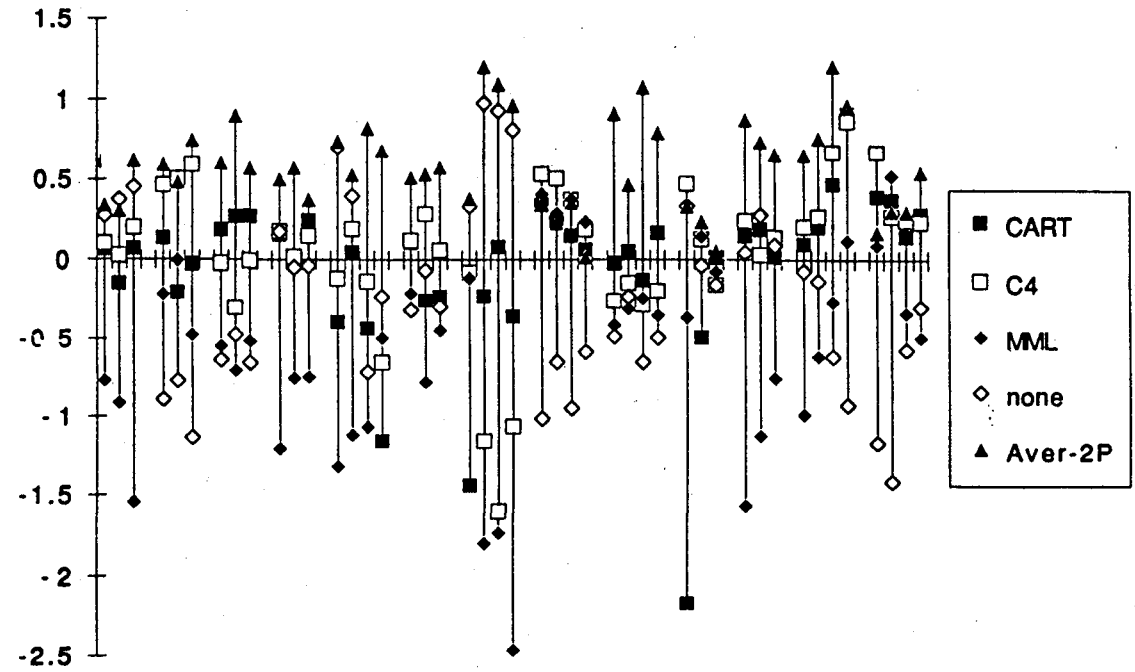


Figure 4: Normalised Accuracies for Methods

Theory Comparison

Bayesian Approximation

The more averaging and smoothing that were done, the better results became. Method sensitive to choice of prior, but very bland prior (constant over structures) worked very well with full averaging.

Minimum Encoding (e.g. MDL)

Very sensitive to choice of prior. Experiments support view that MDL is first-order approximation to full Bayesian. i.e. great gains to be made by averaging, etc.

Resampling methods (e.g. CART)

Have no clear Bayesian interpretation, but appear to overprune so have "implicit" belief favouring simplicity.

Uniform Convergence (basis of most computational learning theory)

No experiments done. (But with small samples probably would have said sample sizes where too small to do anything, and with large samples would have worked very well.)

Extensions

Full approach (growing, smoothing, averaging etc.) extends to:

- Bayesian networks
- variable *n*-gram models (e.g. mixed bi and trigrams)
- regression (function finding), etc.

Missing values (in the attribute vector) and linear combination cut-points of real-valued vectors can be handled using a modified EM-algorithm

Conclusion

- have supported the view that:

  there are other useful approximations but only one normative theory for empirical learning

  e.g. MDL is a first approximation to Bayesian methods

- have illustrated a generic algorithm design strategy for learning structure from data

- more research required in methods (and their quality) of approximating the posterior sum

A Bayesian Method for the Induction of Probabilistic Networks from Data

Edward Herskovits
Section on Medical Informatics
Stanford University

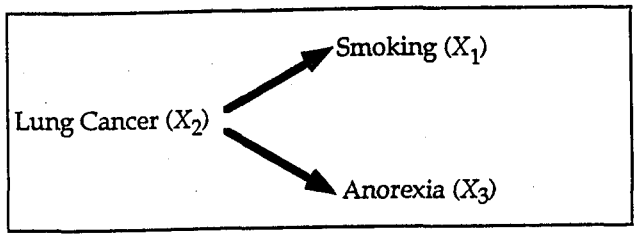
Gregory Cooper
Section of Medical Informatics
University of Pittsburgh

Outline

- Brief introduction to Bayesian belief networks
- An example of hypothesis testing of belief-network structures
- A formula for ranking belief-network structures by their posterior probabilities
- K2: A heuristic procedure that searches for the most probable belief-network structure given a database
- Results using K2
- Other developments
- Open problems

The Bayesian Belief-Network Representation

Belief-network structure B_S:



Belief-network probabilities B_P:

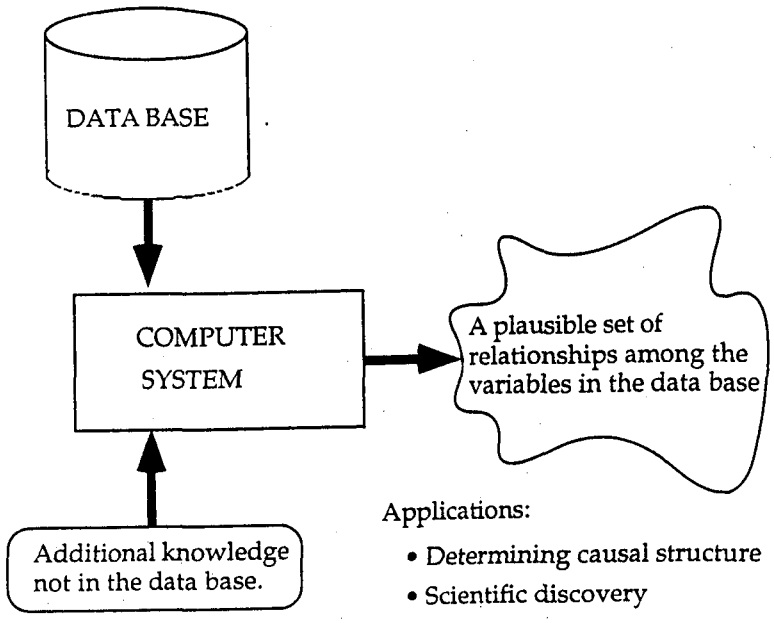
P(Lung Cancer) = 0.01

P(Smoking | Lung Cancer) = 0.75

P(Smoking | no Lung Cancer) = 0.5

P(Anorexia | Lung Cancer) = 0.2

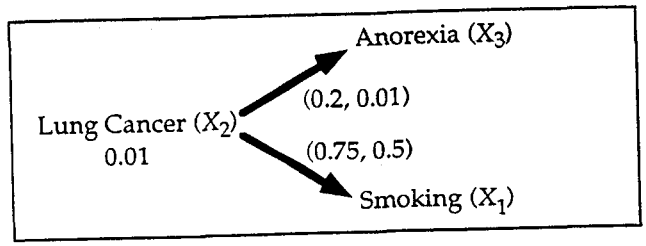
P(Anorexia | no Lung Cancer) = 0.01



Applications:

- Determining causal structure
- Scientific discovery
- Hypothesis testing
- Automated construction of diagnostic systems

Belief network $B = B_P + B_S$:



Properties of belief networks:

- Formally capture the conditional independencies and dependencies among a set of variables.
- Capable of representing any probabilistic distribution over a set of variables.
- An intuitive, graphical model for representing and visualizing probabilistic relationships among variables.
- Algorithms exist for performing probabilistic inference on belief networks.

Examples of probabilistic inference tasks:

P(Lung Cancer | Anorexia)

P(Smoker | Anorexia)

P(Lung Cancer | Anorexia, Smoker)

A database example.
Let D denote this database.

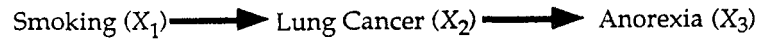
Variable values for each case

case	x_1	x_2	x_3
1	present	absent	absent
2	present	present	present
3	absent	absent	present
4	present	present	present
5	absent	absent	absent
6	absent	present	present
7	present	present	present
8	absent	absent	absent
9	present	present	present
10	absent	absent	absent

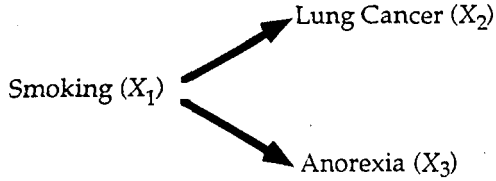
where x_1 denotes *history of smoking*
x_2 denotes *lung cancer*
x_3 denotes *anorexia (severe prolonged loss of appetite)*

Two Belief-Network Structures
(Serving as hypotheses about the dependencies among the three variables)

Hypothesis B_{S1}:



Hypothesis B_{S2}:

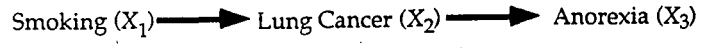


Question: What is the relative likelihood of the two structures, given the data?

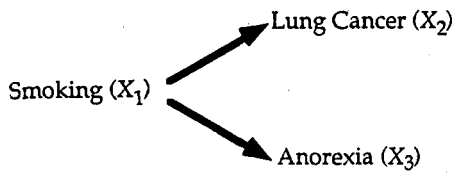
Variable values for each case

case	x_1	x_2	x_3
1	present	absent	absent
2	present	present	present
3	absent	absent	present
4	present	present	present
5	absent	absent	absent
6	absent	present	present
7	present	present	present
8	absent	absent	absent
9	present	present	present
10	absent	absent	absent

Hypothesis B_{S1}:



Hypothesis B_{S2}:



Hypothesis testing

Question: What is the relative likelihood of the hypotheses B_{S1} and B_{S2}?

or restated

What is $\frac{P(B_{S1} \mid D)}{P(B_{S2} \mid D)}$?

$$\frac{P(B_{S1} \mid D)}{P(B_{S2} \mid D)} = \frac{\frac{P(B_{S1}, D)}{P(D)}}{\frac{P(B_{S2}, D)}{P(D)}} = \frac{P(B_{S1}, D)}{P(B_{S2}, D)} \qquad (1)$$

Assumption 1. Model the process that generates a database, as a belief network containing only discrete variables.

The application of Assumption 1 yields:

$$P(B_S, D) = \int_{B_P} P(D \mid B_S, B_P)\, f(B_P \mid B_S)\, P(B_S)\, dB_P, \quad (2)$$

where B_S is the belief-network structure,
B_P is the set of probabilities on B_S,
f is a probability density function over B_P, and
D is the database of cases.

Additional Assumptions

Assumption 2. Cases occur independently, given a belief-network model.

Assumption 3. Cases are complete.

Assumption 4. Probability distributions over the conditional probabilities in a belief network are marginally independent and uniform.

From Assumptions 1 through 4, we can derive a closed-form equation for $P(B_S, D)$:

$$P(B_S, D) = P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \qquad (3)$$

Examples:

$$P(B_{S1}, D) = \frac{1}{25} \frac{5!\,5!}{(5+5+1)!} \frac{1!\,4!}{(1+4+1)!} \frac{4!\,1!}{(4+1+1)!} \frac{0!\,5!}{(0+5+1)!} \frac{4!\,1!}{(4+1+1)!}$$

$$= 8.91 \times 10^{-11}.$$

$$P(B_{S2}, D) = 8.91 \times 10^{-12}.$$

Thus, $\frac{P(B_{S1} \mid D)}{P(B_{S2} \mid D)} = 10.$

Problem: There is a very large search space of belief-network structures.

Number of variables	Number of possible structures
2	3
3	25
5	29,000
10	$\cong 4.2 \times 10^{18}$

K2: A heuristic search for highly probable structures

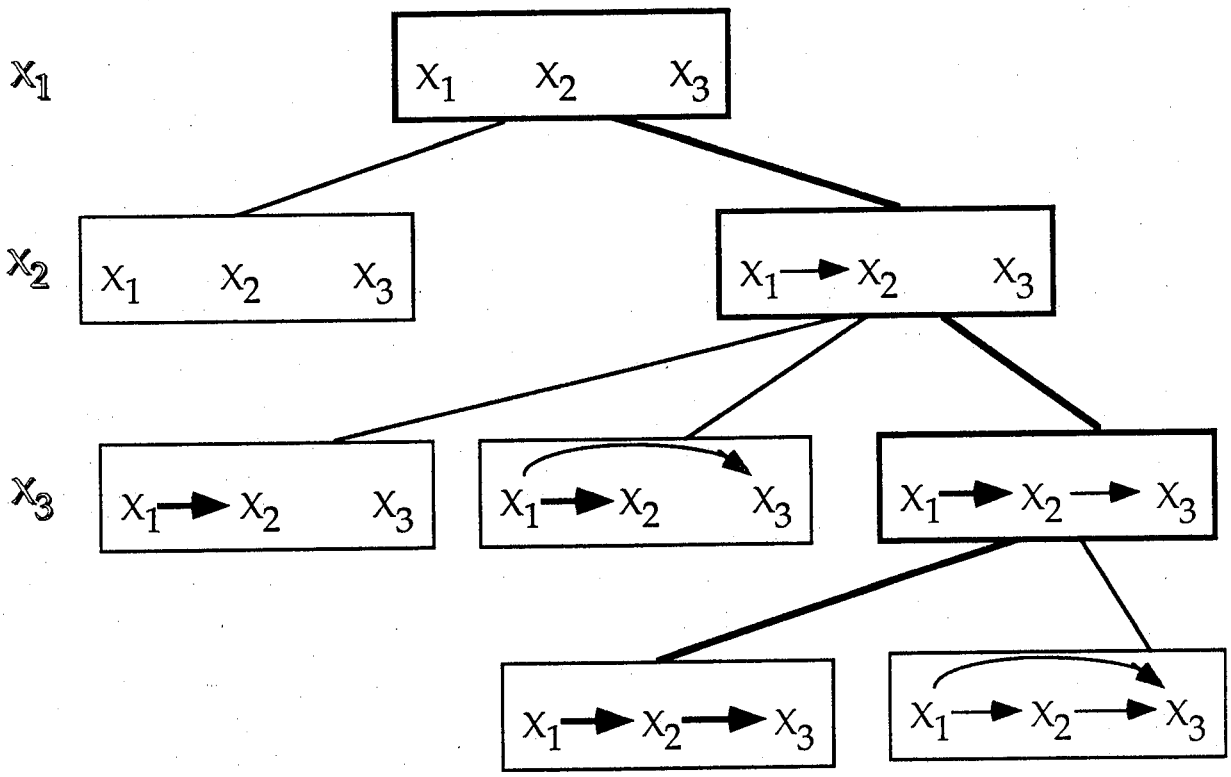
STEP 1: • Assume an ordering on the nodes.

Example: (X_1, X_2, X_3)

• Assume that all structures are equally likely initially.

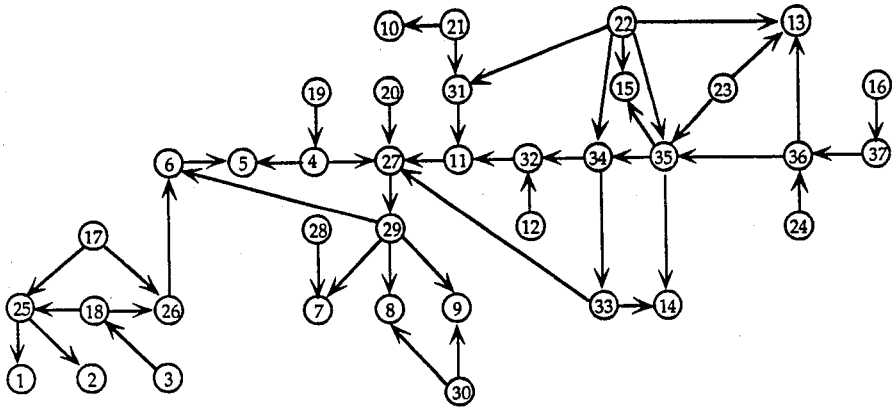
STEP 2: • Apply a greedy search algorithm.

Example:



K2 time complexity: $O(m\ n^4\ r)$ for m cases and n variables, where r is the maximum number of values of any variable.

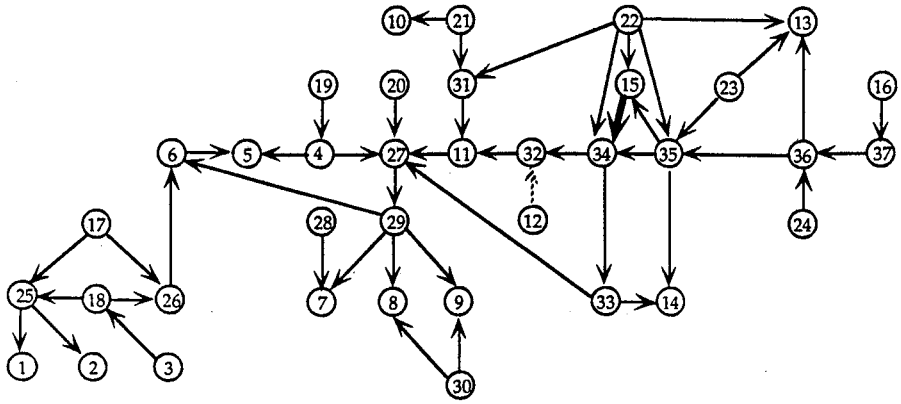
The ALARM Belief-Network Structure



ALARM contains: 37 nodes, 46 arcs, and 2 to 5 values/node.

We generated 3000 cases using ALARM and gave these cases to K2, along with an ordering on the nodes.

Result of Applying K2
to the ALARM-Generated Database



Did not include the arc 12 → 32.
Added the extra arc 15 → 34.

Search time: 5 minutes on a Macintosh II

Results of Applying K2 to Other Databases

K2

Database	# of Cases		# Correct	%Correct
	Training	Evaluation		
LED	200	2300	1667	72
Pathfinder	43	5	5	100
Gyne-Path	130	15	15	100

Kutató

Database	# of Cases		# Correct	%Correct
	Training	Evaluation		
LED	200	2300	1659	72
Pathfinder	43	5	5	100
Gyne-Path	130	15	15	100

Methods also have been developed for

- Using a belief-network structure, plus a database, to perform probabilistic inference,
- Taking a weighted average over multiple belief-network structures to perform probabilistic inference, and
- Handling missing data and hidden variables normatively.

Research topics include:

- Empirical evaluation on a wide variety of databases
- Establishing convergence proofs
- Exploring additional heuristic search methods
- Eliminating the need for a total ordering in K2 by exploring other search methods
- Handling continuous variables
- Developing more efficient search and inference algorithms

Remote Sensing for Ecosystem Monitoring

Chris Hlavka
Ecosystem Science and Technology Branch
NASA Ames Research Center
Moffett Field, CA, 94035

15 January, 1991

Abstract

The Ecosystem Science and Technology Branch at Ames Research Center has been using automated procedures for monitoring ecosystems and renewable resources for the last sixteen years. An overview of our activities and discussions of some recent trends in data analysis will be presented.

Our standard methodologies have been to use both unsupervised and supervised classification techniques applied to imagery on a pixel by pixel basis. Frequently the data sets involve more than one satellite image, involving either multiple overpasses and/or geographical information such as elevation or road networks. Recent work has included exploration of techniques to measure pattern, for example by use of texture measurements on sub-areas of the imagery or measurement of fractal dimension, rather than being limited to per-pixel analysis. Other recent work emphasizes estimates of quantitative variables, the concentration of certain chemicals in forest canopies and unmixing, rather than the categorical determinations made by classification techniques.

OUTLINE OF TALK

- Examples of image processing in the Ecosystem Science and Technology Branch at Ames Research Center:
- Categorical Analysis: Classification using both supervised and unsupervised techniques. (The California Cooperative Remote Sensing Project: 1985 major crops in the Central Valley of California.)
- Analysis of imagery combined with GIS data (The Biospheric Monitoring Disease Prediction Project (Di-Mod): mosquito production).
- Spatial Patterns: Image texture (The Landslide Hazard Assessment Project: mapping land movement).
- Fractal analysis (the Global Biodiversity pilot study: measurement of the fractal dimension of forest boundaries).
- Quantitative Analysis: Analysis of high spectral resolution data (BioGeoChemical research: mapping forest canopy lignin).

Selected References for Remote Sensing For Ecosystem Monitoring

Classification using both supervised and unsupervised techniques (The California Cooperative Remote Sensing Project: 1985 major crops in the Central Valley of California):

- C.A. Hlavka and E.J. Sheffner, The California Cooperative Remote Sensing Project: Final Report, Nasa Tech. Memo. 100073, July 1988.

Analysis of imagery combined with GIS data (The Biospheric Monitoring Disease Prediction Project (Di-Mod): mosquito production):

- Wood, B., B. Washino, L. Beck, M. Pitcairn, D. Roberts, E. Rejmankova, J. Paris, C. Hacker, L. Legters, J. Salute, and P. Sebesta, "Distinguishing High and Low Anopheline Fields Using Remote Sensing and GIS Technology", submitted to *Natural History.*

SPATIAL PATTERNS:

Image texture (The Landslide Hazard Assessment Project: mapping land movement)

- Mckean, J. and S. Buechel; "Remote Sensing of Forested Earthflows", Proceedings of the 1990 U.S. Forest Service Remote Sensing Conference, Arizona, 1990.
- Hlavka, C.A.; "Land-Use Mapping Using Edge Density Texture Measures on Thematic Mapper Simulator Data", *IEEE Transactions on Geoscience and Remote Sensing*, January 1987.

Fractal analysis (the Global Biodiversity pilot study: measurement of the fractal dimension of forest boundaries)

- Mandelbrot, B., *The Fractal Geometry of Nature*, W.H. Freeman and Co., New York, 1977.
- Hlavka, C.A., L.L. Strong, and W.E. Westman, "Remote Sensing of Habitat Fragmentation: An Assessment of Landsat MSS, SPOT, and AVHRR", presented at the Second International Symposium on Advanced Technology in Natural Resource Management, Washington, D.C., November 12 - 15, 1990

QUANTITATIVE ANALYSIS:

Analysis of high spectral resolution data (BioGeoChemical research: mapping forest canopy lignin):

- Card, D.H., D.L. Peterson, P.A. Matson, and J.D. Aber, "Prediction of Leaf Chemistry by the Use of Visible and Near Infrared Reflectance Spectroscopy", *Remote Sensing of Environment*, pp.123-147, 1988.
- Wessman, C.A., J.D. Aber, D.L. Peterson, and J.M. Mellilo; "Remote sensing of canopy chemistry and nitrogen cycling in temperate forest ecosystems", in *Nature*, pp. 154-156, September 1988

Applications of Scale-space Filtering and Labyrinth to Soil Analysis

Deepak Kulkarni and Kevin Thompson

Sterling Federal Systems, NASA Ames Research Center

In this talk, we will present two programs used in the analysis of data produced by a Differential Thermal Analyzer (DTA), a programmable "oven" that heats soil samples at a controlled rate. First, a qualifier program uses scale-space filtering technique to abstract qualitative features from a continuous curve. These qualitative features form a representation of the curve as a structured object. Given this descriptionn, an unsupervised classification program, Labyrinth, creates a hierarchy of classes of minerals in this domain. Labyrinth uses a heuristic measure (category utility) to guide its search for concept hierarchies that will allow prediction of missing information. We will discuss the bottom-up recognition scheme used in Labyrinth and evaluate its performance on actual DTA data.

We shall present a Bayesian method for constructing probabilistic networks from a database of cases. In particular, we focus on constructing Bayesian belief networks. Potential applications include hypothesis testing and automated scientific discovery. We demonstrate how the Bayesian belief networks that are constructed can be used for inference. Applications of such inference include computer-based diagnosis, prediction, and planning. We also discuss the results of a preliminary evaluation of an algorithm for constructing a Bayesian belief network from a database of cases.

This presentation describes material discussed in the technical report, *A Bayesian Method for the Induction of Probabilistic Networks from Data*, by G.F. Cooper and E. Herskovits, available as Knowledge Systems Laboratory Report KSL-91-02, of January 1991 from Medical Computer Science, Stanford University, CA, 94305-5479.

Soil analysis with DTA

- Differential Thermal Analyzer (DTA) is used to heat a soil sample and a reference at a rate defined by a heating program.
- DTA output is a dT verses T graph.

Soil Analysis

- Given: DTA output for a given soil sample

- Find:
- What minerals are present in the sample?

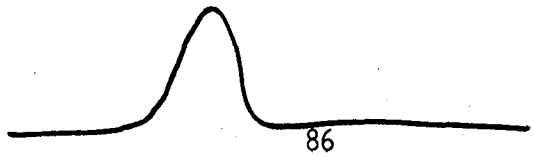
Discovery of Class Hierarchies

- Planetary soils are likely to have unexpected or new minerals.
- Class hierarchies are useful in detecting new minerals.
- Goal: Develop a program to identify class hierarchies in the data.

Diagnostic Features

- Endothermic Reactions

- Exothermic Reactions



Approach for soil analysis

- Extract diagnostic features from the input curve.
- Use a Bayes network classifier to recognize contents.

Approach for identification of classes

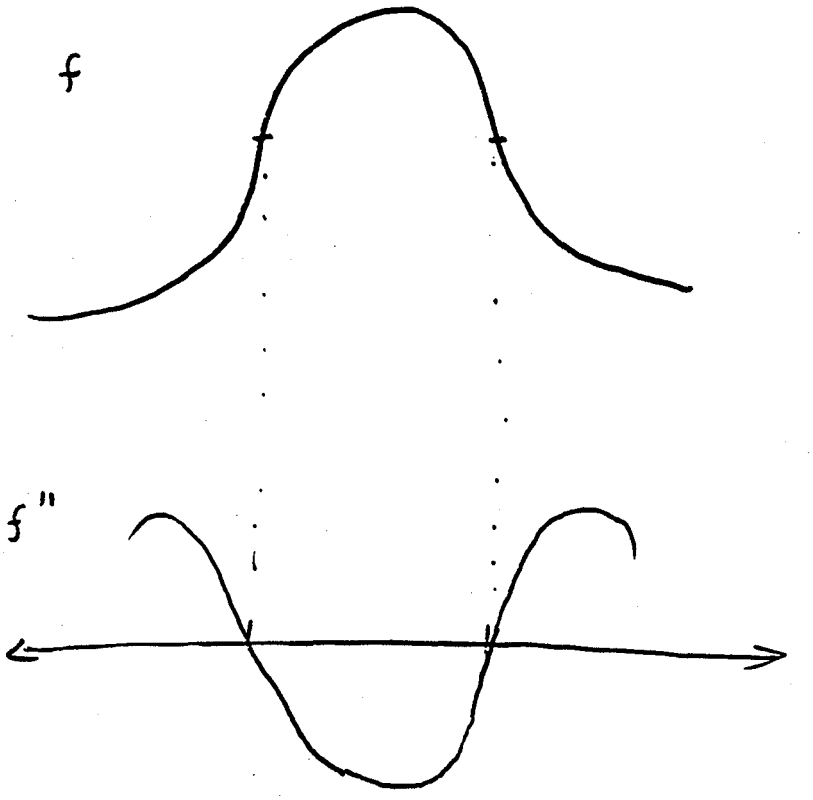
- Extract diagnostic features from the input curve.
- Use LABYRINTH, a program for clustering structured objects, to generate classes.
- Evaluate the utility (or interestingness) of the classes by consulting an expert.

The Qualifier

- Smooth the curve using different gaussian filters.
- Use the zero-crossings in the function to detect edges.
- Use perceptual organization heuristics to detect lines in the scale space graph.
- Use domain specific correlations to generate a probabilistic description.

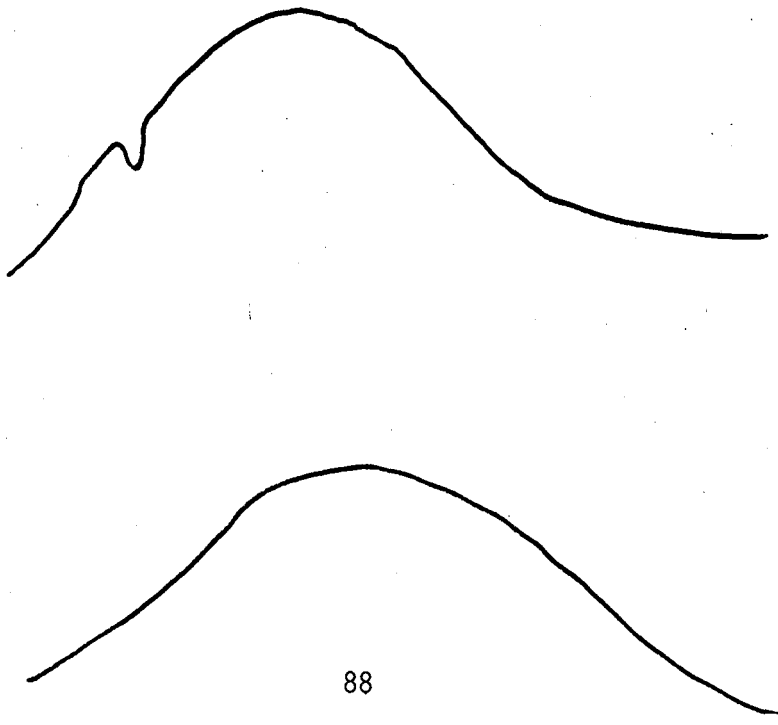
Edge detection

- Inflexion point of an edge corresponds to the zero-crossing in the second derivative.

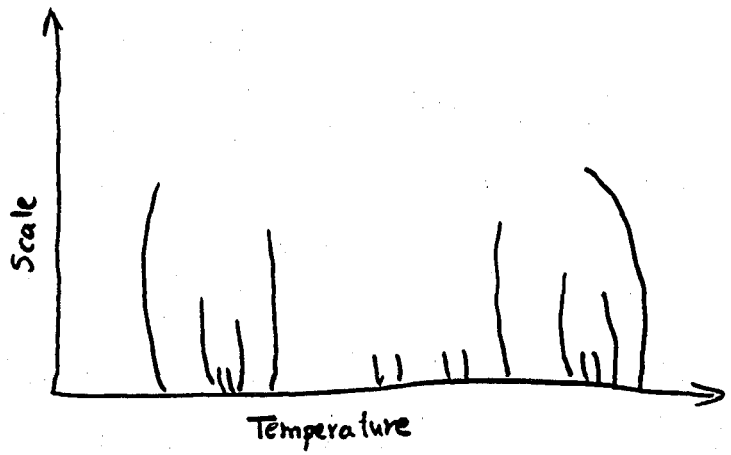


Features at different scales

- Small features vanish when the curve is smoothed.



Scale Space Graph



- Use statistical correlations to associate probability with the features. (e.g. endotherm , positive derivative at first inflexion point and negative derivative at the second inflexion point – .9).

Incremental Concept Formation

We define the task of *incremental concept formation* as

- *Given:* A sequential presentation of instances and their associated descriptions;
- *Find:* Clusterings that group those instances into concepts;
- *Find:* A summary description for each concept;
- *Find:* A hierarchical organization for those concepts.

These concepts are then used to facilitate

- retrieval of concepts based on partial descriptions
- *flexible* prediction of missing information

Concept Formation to Learn about Soil Sample

Goal of the current research: develop a concept formation algorithm that will work in the DTA-GC domain:

- LABYRINTH is a model of concept formation for *structured* objects.
- Extends COBWEB, can learn with objects having varying numbers of components, each with their own description.

Structured Objects

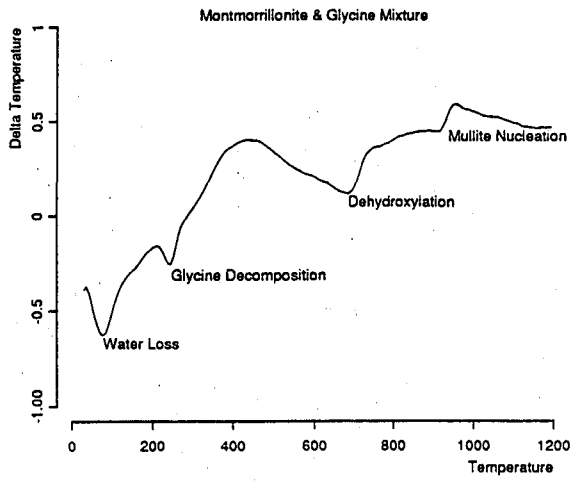
In our framework, a *simple* object is one with an associated set of attributes and their values. A peak extracted from a soil signature:

```
(water-loss (onset-temperature 67.17)
            (peak-width 164.52)
            (peak-temperature 91.76)
            (peak-type endotherm))
```

A *structured* object is one that has components objects; e.g a soil sample:

```
(Montmorillonite
        (water-loss
            (onset-temperature 67.17)
            (peak-width 164.52)
            (peak-temperature 91.76)
            (peak-type endotherm))
        (dehydroxylation
            (onset-temperature 654.57)
            (peak-width 62.97)
            (peak-temperature 690.53)
            (peak-type endotherm))
        (mullite-nucleation
            (onset-temperature 885.71)
            (peak-width 43.67)
            (peak-temperature 929.38)
            (peak-type endotherm)))
```

A Soil Mixture



Concept Representation

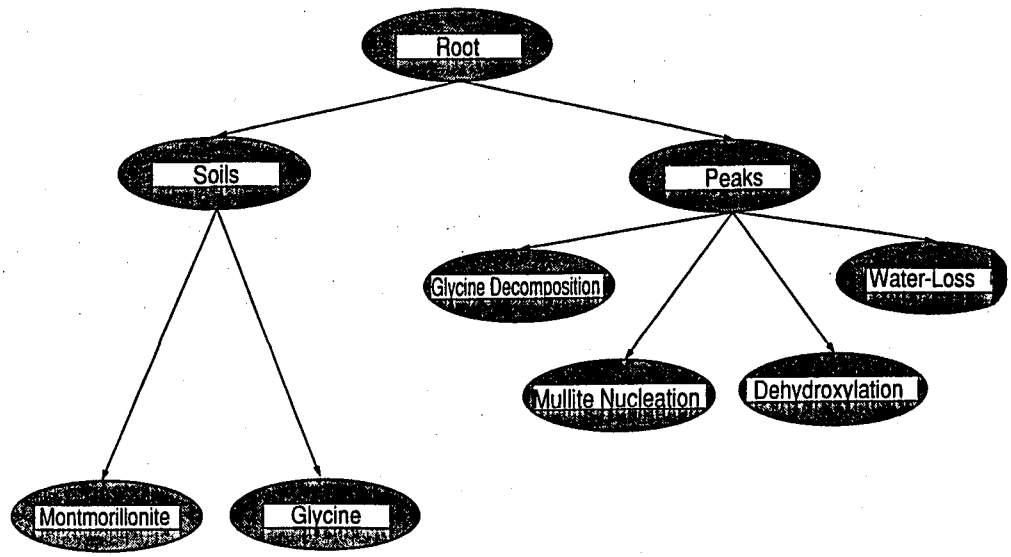
LABYRINTH organizes the structured objects it has encountered into a *probabilistic concept hierarchy*:

- Terminal nodes correspond to either:
  - Observed instances (e.g. soil samples)
  - their components (e.g. peaks).
- Nonterminal nodes represent probabilistic concepts, containing a summary description of the instances (or components) stored below it. A probabilistic concept C_k has:
  - an associated set of attributes A_i and their possible values V_{ij}
  - the conditional probability $P(A_i = V_{ij}|C_k)$ that a value will occur
  - the overall probability of each concept, $P(C_k)$.

LABYRINTH stores concepts for both structured objects and simple objects in the same hierarchy.

- All concepts are indexed by *is-a* links from their parent.
- Two different types of concepts:
  - Simple concepts are stored as in COBWEB, with associated lists of attributes and their values.
  - Structured concepts also have associated values, but here the "values" associated with an "attribute" refer to other nodes in the concept hierarchy.

Memory After Training on Four Soil Samples



Classifying Peaks

- LABYRINTH uses COBWEB to classify each of the component blocks.
- COBWEB returns the name of the most specific concept that the instance matches to an acceptable degree.
- In this case, COBWEB classifies each of the blocks as a member of the WATER-LOSS class.
- LABYRINTH replaces the component description with its label from COBWEB in the instance:

```
(Montmorillonite
        (water-loss  water-loss)
        (dehydroxylation  dehydroxylation)
        (mullite-nucleation  mullite-nucleation))
```

Note that LABYRINTH can now classify the sample as a simple object, with nominal values ("labels") on each attribute.

Extended Example

We begin with memory as shown, after three instances have been seen. We demonstrate how LABYRINTH would incorporate a new instance:

```
(Montmorillonite
        (water-loss
            (onset-temperature 67.17)
            (peak-width 164.52)
            (peak-temperature 91.76)
            (peak-type endotherm))
        (dehydroxylation
            (onset-temperature 654.57)
            (peak-width 62.97)
            (peak-temperature 690.53)
            (peak-type endotherm))
        (mullite-nucleation
            (onset-temperature 885.71)
            (peak-width 43.67)
            (peak-temperature 929.38)
            (peak-type endotherm)))
```

LABYRINTH incorporates this instance by:

1. Classifying each peak based on its four attributes
2. Classifying the soil, based on the "labels" of its component peaks

LABYRINTH Pseudo-Code

```
Input: OBJECT is a composite object, with substructure given.
       ROOT   is the root node of the concept (is-a) hierarchy.
Side effects: Labels OBJECT and all its components with class names.

Procedure Labyrinth(OBJECT, ROOT)

  For each primitive component PRIM of composite object OBJECT,
      Let CONCEPT be Cobweb(PRIM, ROOT);
      Labyrinth'(OBJECT, PRIM, CONCEPT, ROOT).

Procedure Labyrinth'(OBJECT, COMPONENT, CONCEPT, ROOT)

  Label object COMPONENT as an instance of category CONCEPT.
  If COMPONENT is not the top-level object OBJECT,
     Then let CONTAINER be the object of which
                                    COMPONENT is a component.
          If all components of CONTAINER are labeled,
             Then let CONTAINER-CONCEPT be Cobweb'(CONTAINER, ROOT).
                  Labyrinth'(OBJECT, CONTAINER,
                                    CONTAINER-CONCEPT, ROOT).
```

LABYRINTH

- LABYRINTH uses an extended COBWEB to do its sub-tasks
- Works in a "component-first" fashion.
  - first classifies the peaks, "labeling" each one in turn
  - redescribe samples using peak labels as "values"
  - classify these redescribed samples
- The results of previous classifications guide classification of more complex sub-trees of the object
- Each soil sample is stored in terms of other acquired concepts

Overview of COBWEB

COBWEB (Fisher, 1987) incrementally forms concept hierarchies from simple objects.

- represents concepts probabilistically
- tightly integrates classification and learning
- sorts instances from top of tree, at each partition choosing between four operators:
  - Placing the instance in a new, singleton concept
  - Incorporating the instance into an existing concept
  - Merging the two best candidates in partition
  - Splitting the best candidate

COBWEB (continued)

- Uses an evaluation function to determine which operator to apply. Category Utility (Gluck & Corter, 1985) is based on information theory. COBWEB maximizes:

$$\frac{\sum_{k=1}^{K} P(C_k) \sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2}{K}$$

thus favoring high intra-class similarity and high inter-class differences.

For our purposes, COBWEB returns the concept in memory determined to be the best predictor for the parameter object.

Further Potential for Information Extraction from Multispectral Image Data

David Landgrebe
Professor of Electrical Engineering
Purdue University

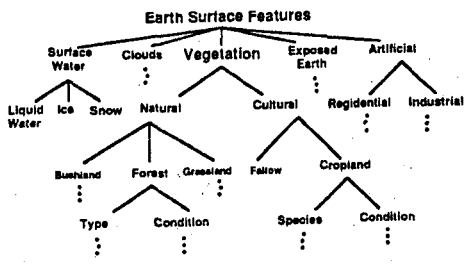
- Focus on high dimensional multispectral imaging devices
- Thinking specifically about HIRIS, MODIS, and the HIRIS/MODIS combination.

Outline of Presentation

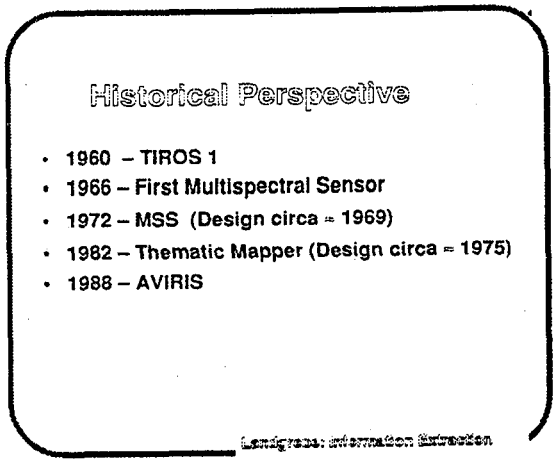
- Background & History
- Basic Axioms
- Ultimate Potential & 2nd order effects

- Common information measures
  - Shannon theory
  - Signal dimensionality
  - # of available themes

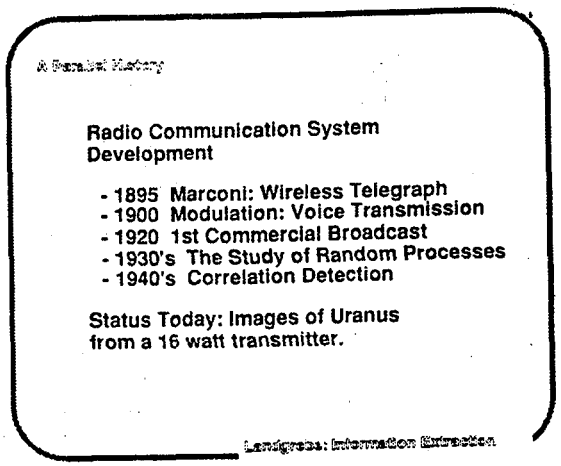
Information as a Taxonomy



- Taxonomies are used whenever the information complexity becomes too great
- For example, in classical science Plant or Animal species,
- For any given Earth Science problem one is free to make up one's own taxonomy
- Each Problem has its own
- A historical perspective may be useful in aiding insight



- In the early 60's one of the questions that motivated Earth observational remote sensing was the search for life on Mars, and thus what resolution is needed
- The multispectral approach soon emerged
- Early field and airborne sensors had much greater spectral resolution than could be built in spacecraft



- The key advancement after initial establishment of the technology came from suitably modeling both signal and noise
- Information potential not apparent from casual look at the signal
- Next turn to some fundamentals of information extraction

Fundamentals

Axiom 1. The availability of information

In remote sensing,
information is available at the aperture of a sensor
based upon the electromagnetic fields
emanating from the surface and arriving there,
and in particular via the,

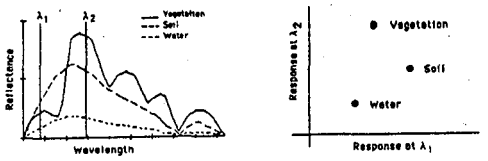
- Spectral,
- Spatial, and
- Temporal

Variations of those fields.

Landgrebe: Information Extraction

- Primary emphasis has been placed upon spectral variations
- Spatial resolution available and themes desired not well mated
- Temporal variations difficult to measure

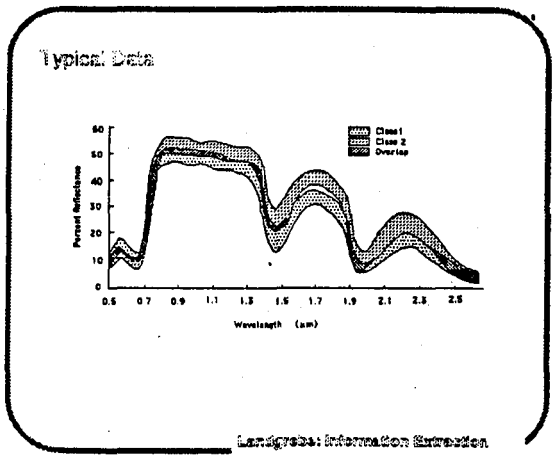
Multidimensional Space



Landgrebe: Information Extraction

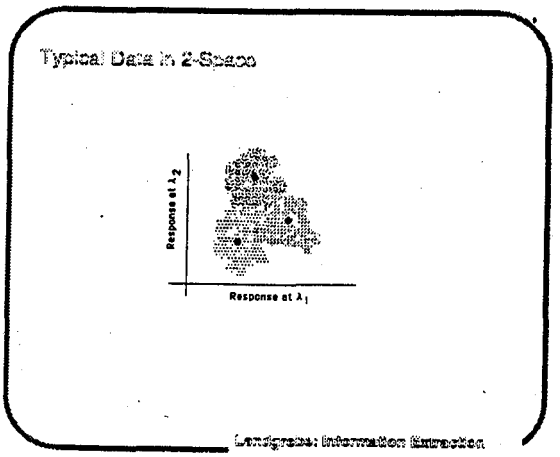
Multispectral Data in Multivariant Space

- Most general representation of spectra
- Can handle, for example, the case of identification by absorption bands
- But does this approach adequately model nature?



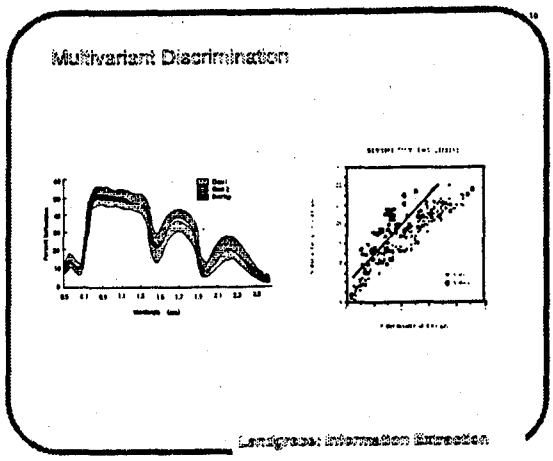
Typical mutispectral data

- High Resolution Data
- Laboratory Data, thus many of the vagaries of in situ observation not present.



Typical data in multivariant space

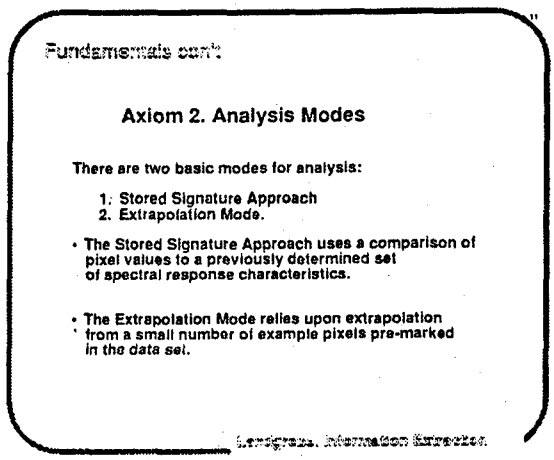
- Data not a point for a class but a distribution
- Must learn to deal with classes that are not completely separable



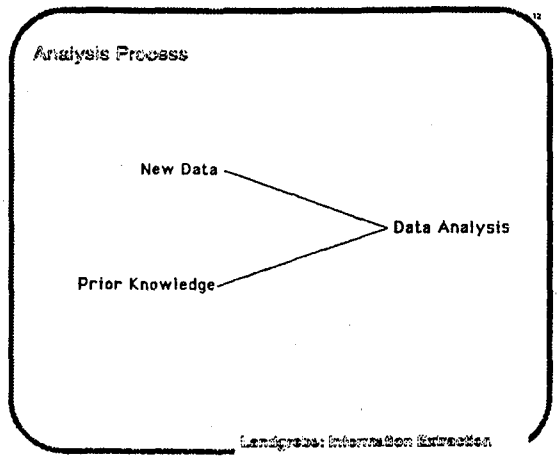
- Separability in 0.7 μm region not apparent from spectral curves.
- Is apparent in multivariant space (more later)

Conclusions:

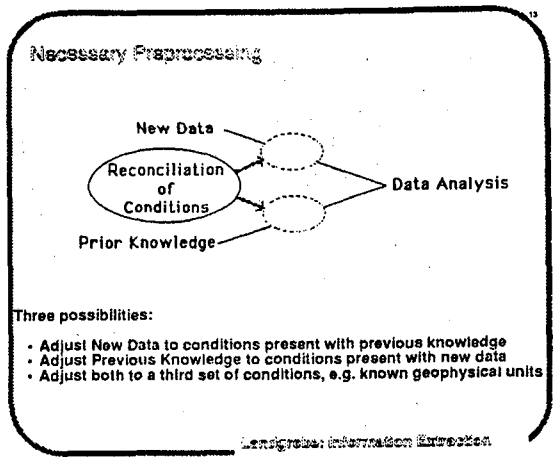
- Straightforward inspection of the spectral response does not reveal all of the potential for discrimination
- In higher dimensional cases the discrimination potential cannot be humanly observed even in multivariant space



- Stored Signature Approach is the most straightforward
- Extrapolation Mode requires manual labelling of samples within the data set. However, for that effort, one normalizes out many of the observation variables, such as atmospheric and goniometric effects.



- **The critical problem of the analysis process stems from the above merging of data, which is necessarily inherent in the analysis process**



- **Usual approach is to do the former of these - requires much processing**
- **The second of these is inherent in the extrapolation mode**
- **The third requires the most processing of all**

Further Fundamentals

Axiom 3. Absolute & Relative Classification

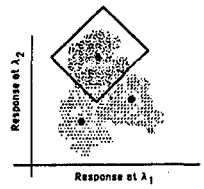
Classification, i.e. discrimination between classes, on both an *absolute* and a *relative* basis are useful.

- Absolute Classification implies identification of a given class of material wherever it occurs in the data set without regard to any other class which may be present.
- Relative Classification implies discrimination between classes, i.e. deciding in favor of a given class after having considered all possible classes in the scene.

Landgrebe: Information Extraction

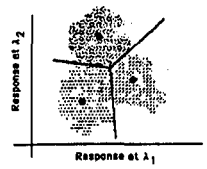
- **Absolute is the most natural and easiest to implement**
- **Relative provides greater accuracy and greater penetration into the information tree**

Absolute & Relative Classification



Absolute Classifier

- Easy to Implement

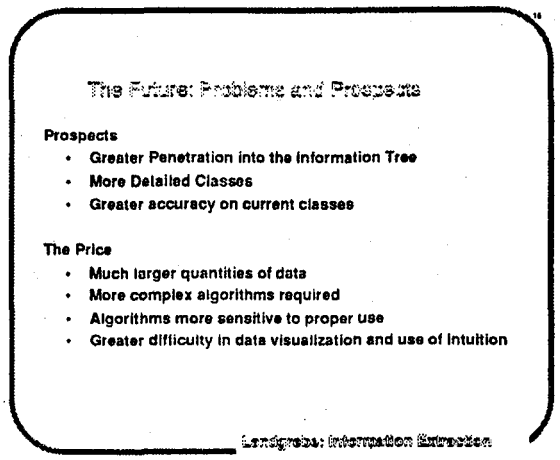


Relative Classifier

- Greater Accuracy & More subtle classes

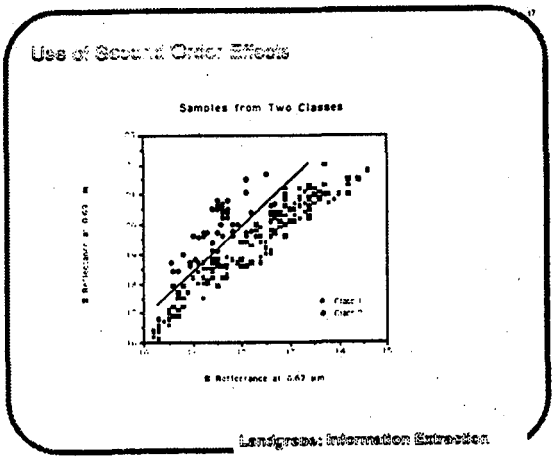
Landgrebe: Information Extraction

- In the N-space this appears as above.
- In the Absolute Case, one uses only information about the class of interest to locate the decision boundary
- In the Relative Case, one uses information about all classes to locate the decision boundaries, even if one is only interested in a single class



Simple algorithms still useful, but full value of the new data will require more powerful algorithms.

- The more complex data requires more complex modeling and analysis schemes for full potential
- The case for use of second order effects
- To illustrate, Return to the 2-class example



- Note that it is the second order statistical effects, i.e., the covariance, which enabled discrimination in this case

- Had the data had the same mean and variance but been uncorrelated between these two bands, discrimination would have been poor

A STRAGEDY FOR

LARGE-AREA LAND COVER CHARACTERIZATION

By Thomas R. Loveland[1/], James W. Merchant[2/], and Donald O. Ohlen[3/]

ABSTRACT

Global climate change research requires spatial information describing the distribution and composition of land surface phenomena over very large areas. The data and methods for producing the needed data bases must deal with a variety of technical issues, including (1) problems related to the large geographic area covered and the inherent seasonal, ecological, and cultural variations; (2) handling the large volumes of data involved; and (3) data of varying quality. In addition, the results, rather than the classification schemes, must provide the flexibility for descriptions of the landscape based on a range of surface characteristics, such as vegetation types, land cover components, surface albedo, leaf area index.

An investigation is underway to test a strategy for large-area land cover characterization by using National Oceanic and Atmospheric Administration 1.1-kilometer Advanced Very High Resolution Radiometer imagery and ancillary spatial data (elevation, climate, historical land use/land cover, and ecological regions) for the conterminous United States. The prototype strategy involves unsupervised digital classification of time series vegetation index and brightness data spanning 1990, followed by refinement of the image classification by using ancillary spatial data with location-based logical rules to produce a stratification of land surface features. When combined with ancillary land data and satellite-derived geophysical parameters, the resulting land cover characteristics data base can be used to tailor the data to the requirements of specific application.

1/ U.S. Geological Survey

2/ University of Nebraska-Lincoln, Center for Advanced Land Management Information Technologies.

3/ TGS Technology, Inc. Work performed under U.S. Geological Survey contract 14-08-0001-22521.

CONTERMINOUS U.S. LAND CHARACTERIZATION DESIGN CONSIDERATIONS

- Repeatable over large areas (continental, global)
- Significant seasonal, ecological, and cultural variations
- Large data volume
- Varying data quality
- Flexible results that are not application-specific

FILE NAME: SAB

USGS/EDC
1-02-91

CONTERMINOUS U.S. LAND CHARACTERIZATION POTENTIAL APPLICATIONS

- LAND COVER MAPPING
  - — Land cover components
  - — Vegetation components
- ECOLOGICAL REGIONALIZATION
  - — Land use/cover components
  - — Vegetation components
  - — Selected biophysical parameters
- CLIMATE AND RESOURCE MODELING
  - — Biophysical parameters
  - — Land use/cover components
  - — Vegetation components

FILE NAME: SAB

USGS/EDC
1-02-91

CONTERMINOUS U.S. LAND CHARACTERIZATION MULTISOURCE DATA BASE

- TIME-SERIES ADVANCED VERY HIGH RESOLUTION RADIOMETER (AVHRR) DATA
  - — Normalized Difference Vegetation Index (NDVI)
  - — Brightness
- TERRAIN DATA
  - — Elevation
  - — Slope and Aspect
  - — Solar Illumination
- CLIMATE DATA
  - — Minimum/Maximum Monthly Temperature
  - — Monthly Precipitation
  - — Frost-Free Period
- ECOLOGICAL REGIONS (EPA/OMERNIK)
- MAJOR LAND RESOURCE AREAS
- USGS LAND USE/LAND COVER

FILE NAME: SAB

USGS/EDC
1-02-91

VEGETATION CONDITION

- Initial greenup
- Magnitude of greenness
- Duration of greenness
- Seasonal changes
- Year to year changes

FILE NAME: SAB

USGS/EDC
11 - 15 - 89

NORMALIZED DIFFERENCE VEGETATION INDEX

A ratio of near-infrared and visible radiances

$$\frac{\text{NEAR INFRARED (CH.2)} - \text{VISIBLE (CH.1)}}{\text{NEAR INFRARED (CH.2)} + \text{VISIBLE (CH.1)}}$$

where: possible data range is -1.0 to 1.0
- — negative values indicate non-vegetative surfaces (water, clouds, etc.)
- — positive values indicate vegetative surfaces
- — high positive values indicate high vegetation density and or vigor

FILE NAME: SAB

USGS/EDC
11 - 15 - 89

AVHRR MEASUREMENTS OF LAND SURFACE FEATURES

Stratification of NDVI response to broad scene components as measured from NOAA-7 [1]

COVER TYPE	PLANETARY ALBEDO CHANNEL 1	PLANETARY ALBEDO CHANNEL 2	NDVI
Dense green-leaf vegetation	.050	.150	.500
Medium green-leaf vegetation	.080	.110	.140
Light green-leaf vegetation	.100	.120	.090
Bare soil	.269	.283	.025
Clouds (opaque)	.227	.228	.002
Snow and ice	.375	.342	-.046
Water	.022	.013	-.257

[1] Holben, B. N. "Characteristics of maximum value composite images from temporal AVHRR data". International Journal of Remote Sensing, Vol 7, No. 11, November, 1986.

FILE NAME: SAB

USGS/EDC
11-15-90

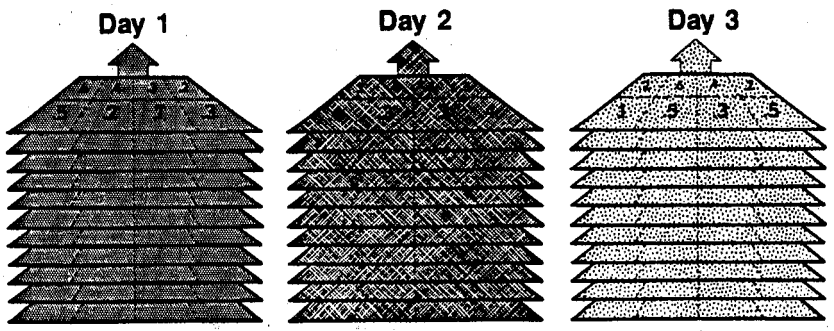
CONTERMINOUS U.S. LAND CHARACTERIZATION PROJECT OBJECTIVES

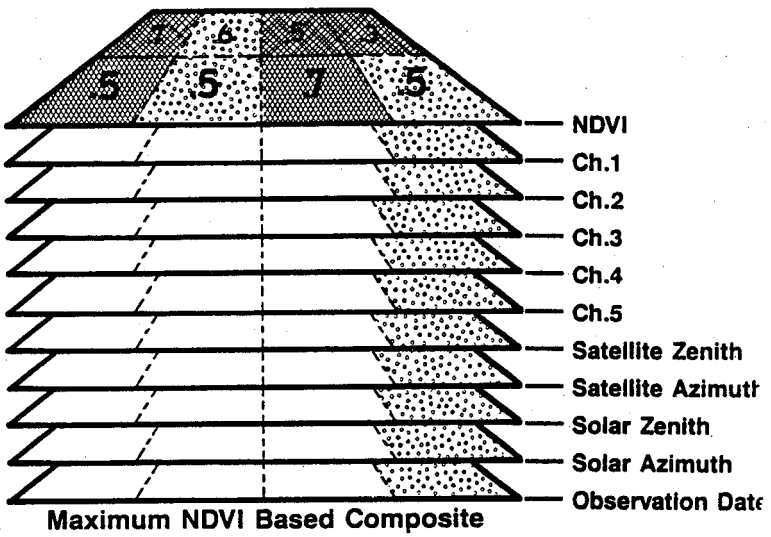
- Develop methods for producing large area land cover characteristics data bases.
- Create a "prototype" land cover characteristics data base for the conterminous U.S. that can be evaluated and scrutinized by the scientific community in order to understand the utility and limitations of the data base for global climate research and other problems.

FILE NAME: SAB

USGS/EDC
1-02-91

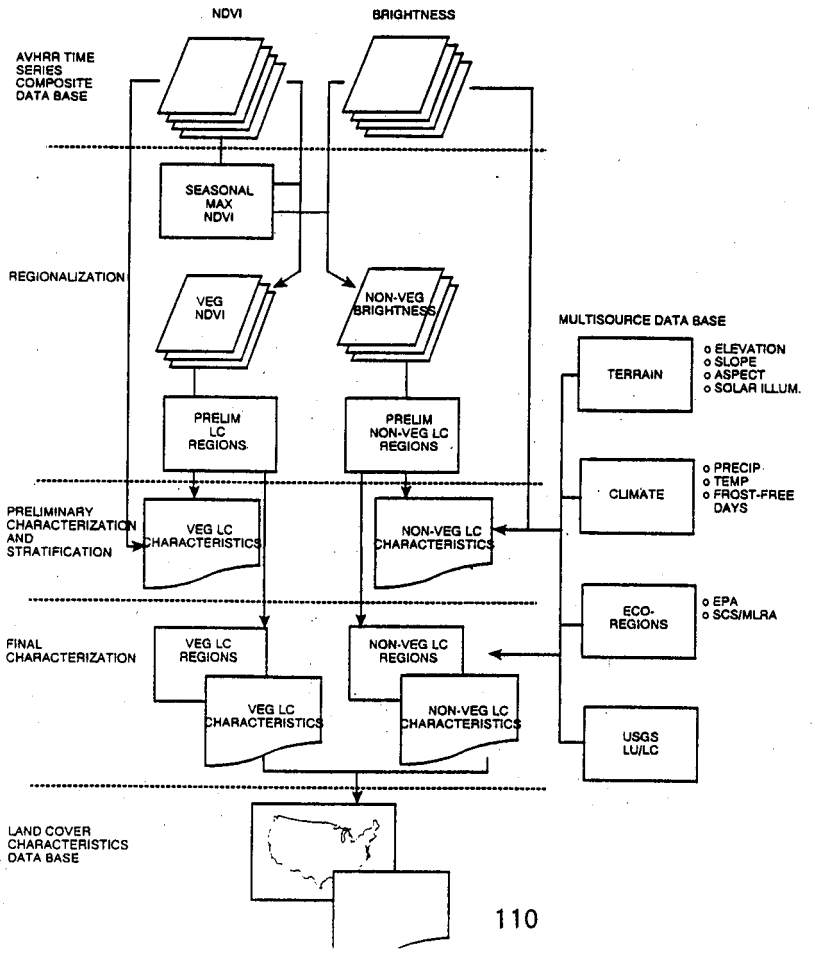
Maximum NDVI Image Compositing

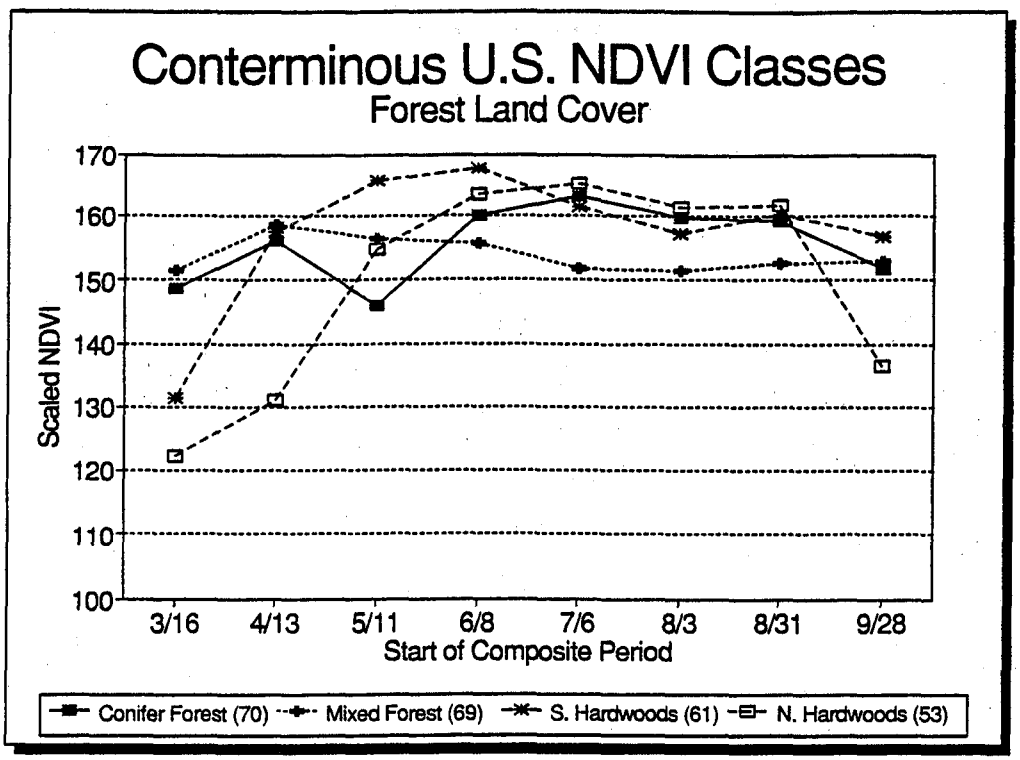


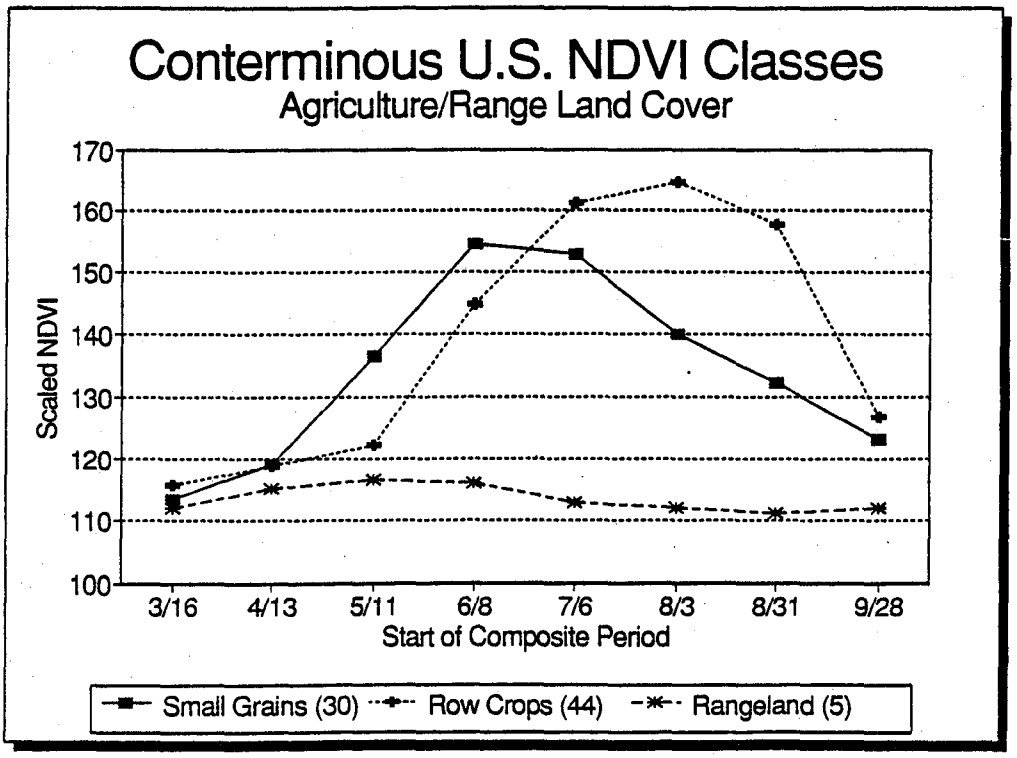


Maximum NDVI Based Composite

Conterminous U.S. Land Characterization




Conterminous U.S. NDVI Classes
Forest Land Cover
Scaled NDVI
170
160
150
140
130
120
110
100
3/16
4/13
5/11
6/8
7/6
8/3
8/31
9/28
Start of Composite Period
Conifer Forest (70)
Mixed Forest (69)
S. Hardwoods (61)
N. Hardwoods (53)


Conterminous U.S. NDVI Classes
Agriculture/Range Land Cover
Scaled NDVI
170
160
150
140
130
120
110
100
3/16
4/13
5/11
6/8
7/6
8/3
8/31
9/28
Start of Composite Period
Small Grains (30)
Row Crops (44)
Rangeland (5)

NDVI Characteristics

CLASS	3/16	4/13	5/11	6/8	7/6	8/3	8/31	9/28
5	111.72	115.05	116.67	115.92	112.65	111.77	110.97	111.85
30	113.34	118.97	136.47	154.65	152.88	139.90	132.39	122.89
44	115.61	118.70	122.29	144.80	161.28	164.64	157.96	126.84
53	122.24	131.08	154.68	163.28	165.16	161.59	161.90	136.47
61	131.42	157.51	165.72	167.49	161.40	157.38	160.39	156.91
69	151.52	158.73	156.59	155.68	151.94	151.54	152.62	152.94
70	148.80	156.12	145.91	160.16	163.13	159.91	159.21	151.72

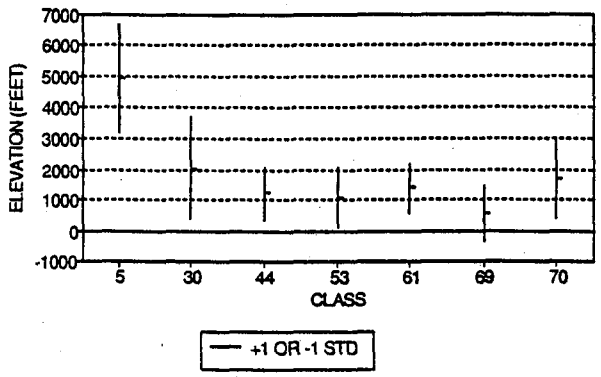
LAND COVER/VEGETATION Characteristics

CLASS	LAND COVER	ANDERSON II	VEGETATION
5	OPEN SHRUBLAND	SHRUB AND BRUSH RANGELAND	SALTBRUSH/GREASEWOOD/SAGE
30	SMALL GRAINS	CROPLAND AND PASTURE	SPRING WHEAT
44	ROW CROPS	CROPLAND AND PASTURE	CORN/SOYBEANS
53	DECIDUOUS FOREST	DECIDUOUS FOREST LAND	MAPLE/BIRCH
61	DECIDUOUS FOREST	DECIDUOUS FOREST LAND	OAK/HICKORY
69	MIXED FOREST	MIXED FOREST LAND	OAK/HICKORY/PINE
70	CLOSED CONIFER FOR	EVERGREEN FOREST LAND	SPRUCE/CEDAR/HEMLOCK/FIR

ELEVATION Characteristics

CLASS	MEAN	STD	MAX	MEDIAN
5	4935	1751	13300	5296
30	2043	1663	12021	1600
44	1197	862	11631	1047
53	1088	1011	11572	1111
61	1367	830	6534	1142
69	558	912	7970	260
70	1681	1326	7419	1329

CONTERMINOUS U.S. LAND CHARACTERISTICS
ELEVATION - CLASS RELATIONSHIPS



CLASS	COVER	MEAN	STD	MAX	MEDIAN
5	RANGELAND	4935	1751	13300	5296
30	SMALL GRAINS	2043	1663	12021	1600
44	ROW CROPS	1197	862	11631	1047
53	N. HARDWOODS	1088	1011	11572	1111
61	S. HARDWOODS	1367	830	6534	1142
69	MIXED FOREST	558	912	7970	260
70	CONIFER	1681	1326	7419	1329

ECOLOGICAL REGIONS/NDVI CLASS RELATIONSHIPS

CLASS	PERCENT	REGION	CHARACTERISTIC COVER TYPES
5	26.7%	N. BASIN & RANGE	DESERT SHRUBLAND
	15.9%	COLORADO PLATEAU	WOODLANDS, DESERT SHRUBLAND
	14.6%	SNAKE RIVER BASIN	DESERT SHRUBLAND
	57.2%		
30	30.2%	N. GLACIATED PLAINS	CROPLAND
	20.7%	N.W. GLACIATED PLAINS	CROPLAND, PASTURE
	19.1%	RED RIVER VALLEY	CROPLAND
	70.0%		
44	45.6%	WESTERN CORN BELT	CROPLAND
	26.6%	CENTRAL CORN BELT	CROPLAND
	7.2%	EASTERN CORN BELT	CROPLAND
	79.4%		
53	75.4%	N. LAKES & FOREST	HARDWOODS (MAPLE, BIRCH, FIR)
	5.2%	N.C. HARDWOOD FOREST	HARDWOODS (MAPLE, BASSWOOD)
	4.3%	N.E. HIGHLANDS	HARDWOODS/SPRUCE
	84.9%		
61	20.6%	CENTRAL APPALACHIAN	APP. OAK, MESOPHYTIC FOREST
	19.1%	OZARK HIGHLANDS	OAK/HICKORY
	12.4%	INTERIOR PLATEAU	OAK/HICKORY
	52.1%		
69	32.4%	SOUTEASTERN PLAINS	MESOPHYTIC FOREST
	24.7%	SOUTH CENTRAL PLAINS	OAK/HICKORY/PINE
	7.0%	MISS. VALLEY PLAINS	OAK/HICKORY/PINE
	64.1%		
70	33.2%	COAST RANGE	SPRUCE/CEDAR/HEMLOCK
	28.1%	CASCADES	FIR/HEMLOCK/SPRUCE
	15.7%	PUGET LOWLAND	CEDAR/HEMLOCK/FIR
	77.0%		

A NEURAL-BASED APPROACH TO EXTRACTING TEMPORAL DATA FROM A LARGE-SCALE NUCLEAR POWER DATABASE

Alianna J. Maren (1,2), Awatef Gacem (1), and Robert E. Uhrig (1)

(1) Dept. of Nuclear Engineering
The University of Tennessee
Knoxville, TN 37996-2300

(2) The University of Tennessee Space Institute
Tullahoma, TN 37388-8897

A large-scale data base, the SCSS, symbolically encodes sequences of occurences leading to reported events in nuclear power plants. There are several major complexities in identifying and extracting implicit information from such a large-scale database, although there would be substantial practical advantages to finding an automated means of extracting useful temporal information. This task is not amenable to the usual algorithmic and/or symbolic approaches. We have developed a bilayer self-organizing topology-preserving map approach to encoding the relationships between symbolic representations of causes and effects. This mapping is useful, as one cause may lead to multiple types of effects, and one effect may be associated with more than one cause. We are exploring different means of embedding temporally associative relationships between different causes and effects. This involves modifying the basic Kohonen network to include temporally-persistant neural activations and special connectivities between neurons, both inter- and intra-layer. We believe that this novel neural-based architecture will lead to the ability to extract strings of temporally associated occurences given minimal initial stimulus from a trained network. This will be useful in identifying "typical" sequences of occurences which are likely to lead to a reported event.

APPLICATION OF NEURAL NETWORKS TO DATABASE MINING

Awatef Gacem
Alianna Maren and Robert Uhrig
University of Tennessee
November 14, 1990

Neural networks have been applied to many applications in Nuclear Engineering. Examples of applications are fault diagnosis, and signal validation. Most of these applications are based on the Multi Layer Perceptron (MLP) with backpropagation training. A new application based on another neural network, the Kohonen Self-Organizing neural network is the mining of a nuclear database: the Sequential Coding Search System (SCSS). Mining a database is extracting information that is hidden. The SCSS is an Impressive database formed by about 30000 records of LER's. An LER is a description in natural language of an incident. In order to be processed automatically, the LER's are transformed into symbolic description by the Oak Ridge National Laboratory. The SCSS was built to allow the Nuclear Safety Analyss to benefit from the Knowledge gained from the past experience.

PROBLEM STATEMENT

A large amount of knowledge is stored in the SCSS database. The knowledge in this database is either explicit or implicit.

1. The explicit knowledge is formed by the records of the incidents. It is exploited to provide statistical information about the incidents.

2. The implicit knowledge is formed by patterns and relationships or by an explicit information that is missing. This knowledge is not exploited and needs to be unraveled.

TECHNICAL OBJECTIVE

The purpose of our research is to develop a technique to extract implicit knowledge from the data explicitly stored in the large Nuclear Database: the SCSS.

DESCRIPTION OF THE SCSS

The Sequential Coding Search System (SCSS) is a large temporal database developed by ORNL for the NRC.
In the SCSS are stored records of all incidents since 1980 in all nuclear plants in the USA. About thirty thousands events were recorded in the past decade.
The purpose of the SCSS is the analysis of trends and patterns.

DESCRIPTION OF THE CODED STEP MATRIX

The coded step matrix encodes an event which is a chronologically ordered sequence of occurrences.
Each line of the encoding matrix is an occurrence.

An occurrence is defined as a causal relationship (cause/effect relation), and is characterized by several facts (cause, primary system, component, ...)
Each fact is identified by a symbolic variable formed by one to four alphabetical letters.

FIGURE 2. AN EXAMPLE OF A SINGLE LER DATABASE RECORD

Header Information

DOCKET	YEAR	LER NUMBER	REVISION	DCS NUMBER	NSIC	EVENT DATE
293	1984	005	0	8405080272	189610	4-4-1984

Comments

COMMENTS
VALVES MODEL #7567F. STEP 2: COMP XVZ = PILOT VALVE.

Docket Information

DOCKET: 293 PILGRIM 1 TYPE: BWR
REGION: 1 NSSS: GE
ARCHITECTURAL ENGINEER: BECH
FACILITY OPERATOR: BOSTON EDISON CO.
SYMBOL: BEC

Watch-List Codes

WATCH-LIST CODES FOR THIS LER ARE:
913 UPDATE NEEDED

Reportability Codes

REPORTABILITY CODES FOR THIS LER ARE:
10 10 CFR 50.73(a)(2)(i): Shutdowns or technical specification violations.

Reference LERs

REFERENCE LERS:

1 293/81-062

Coded Step Matrix

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF
1	0		PH	BR		VLVS		2	1		1	M	TR	I	DC
2	1		RC	BR		XVZ	T020	1	1		1	A	TR	I	KB
3	2		RC	BR		ORVZ	T020	2	1		1	A	TR	I	AL
4				XX								H	XX		YC
5				YY								N	N		YC

Abstract

ABSTRACT
POWER LEVEL - 000%. ON 4/4/84, DURING A REFUELING OUTAGE, THE MAINTENANCE DEPARTMENT WAS NOTIFIED BY WYLE LABORATORIES THAT THE PILOT VALVES ON TWO OF THE TARGET ROCK TWO-STAGE SAFETY RELIEF VALVES (S/RV'S) DID NOT LIFT WITHIN SPECIFICATION WHEN DIAGNOSTICALLY TESTED IN THE AS-FOUND CONDITION. THIS IS CONTRARY TO THE REQUIREMENTS OF THE INTENT OF PNPS TECH SPEC 2.2.B WHICH REQUIRED THE S/RV'S TO LIFT AT 1095 PSI PLUS OR MINUS 11 PSI. THE MOST PROBABLE CAUSE OF THE SAFETY RELIEF VALVES NOT LIFTING HAS INITIALLY BEEN DETERMINED TO BE STUCK PILOT VALVES. DETERMININATION OF ROOT CAUSE AND CORRECTIVE ACTION IS PENDING FURTHER ANALYSIS AND TESTING.

Table. EXAMPLES OF CAUSE/EFFECT GENERIC CODES

Code	Generic Description
A	Assembly adjustment
B	Leakage
C	Mechanical
D	Mechanistic
Q	Electrical
K	Functional
L	Instrumentation
N	Ambient condition
R	Resultant
S	Human factor
T	Human factor
V	Root cause
W	Parameter of attuation
G	Parameter of actuation
Y	Informative

Q / L

Q / D

S / T

R / Q

CONSIDERATIONS IN SELECTING A NEURAL NETWORK

The selection of a type of the neural network is influenced by the following requirements:

- ability to provide an abstract representation of a highly multidimensional and variable data.
- ability to process symbolic information.
- ability to process temporal information (This feature will be used to classify sequences of cause/effect relations).

This leads to the selection of the Kohonen' Self-organizing features map.

THE KOHONEN SELF-ORGANIZING FEATURES MAP

The KSOFM algorithm is an iterative implementation of the *K-means* clustering algorithm.
It performs a mapping of a higher dimensional input space into a two-dimensional grid of artificial neurons without supervision.
The mapping is generated in such a way that topologically close neurons are sensitive to inputs that are physically similar.
This mapping preserves also the statistical properties of the input data.

USE OF A NEURAL NETWORK TO EXTRACT IMPLICIT INFORMATION FROM A LARGE DATABASE

The proposed neural network-based technique used to extract implicit information from the SCSS is a two-phase technique:

1. The first phase is to compress the explicitly stored data in the SCSS by establishing a mapping from the cause space to the effect space to encode all possible cause/effect relations.

2. The second phase is to use the Causal mapping to generate all possible sequences of cause/effect relations, and identify exemplars of sequences of cause/effect relations that will be used in classification.

SIMULATIONS

- Nuclear plants built by Westinghouse.
- A subset of incidents where a reactor trip was preceded by an electrical problem.
- A cause/effect relation is represented by two facts: the cause and the effect.
- Only the first letter of the cause/effect generic code was used.
- Gray coding was used as a numeric representation of the symbolic data.

2-D MAPPING OF CAUSE/EFFECT RELATIONS

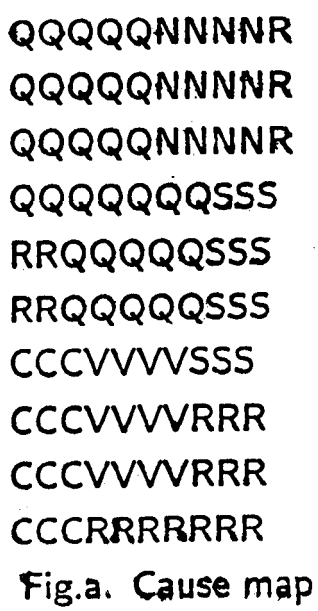


Fig.a. Cause map

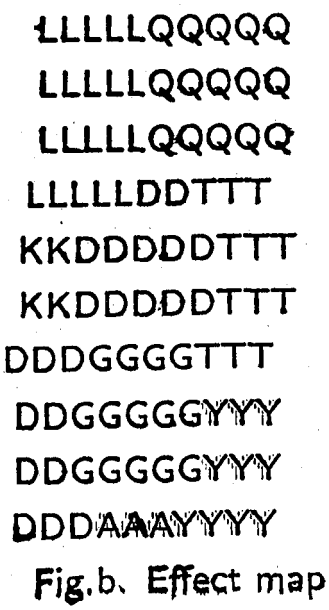


Fig.b. Effect map

RESULTS

The examination of the maps shows the following results:

1. Formation of clusters based on neighborhood relations.

2. A mapping that preserves the probability distribution of the data

SUMMARY OF THE RESULTS

1. The reduction of a large amount of training data without loss of information.

2. The visualization of the relationships among causes or among effects.

3. The mapping between the cause space and effect space which solves the combinatorially explosive problem.

4. A useful representation for processing of sequences of cause/effect relations.

5. A proof that the KSOFM although developed for processing numeric data can operate with symbolic data.

FUTURE WORK

1. Use a multilayer system to represent all the information available about an occurrence (cause/effect relation).

2. Use a cooperative learning method to train the whole network with sequences of occurrences.

3. Identify all possible patterns of sequences.

4. Define a metric to classify the incidents.

CONCLUSION

The development of a methodology to extract implicit knowledge not only will allow a better exploitation of the database, but it may also lead to an improvement of the reliability of the data stored. That is, it may allow to inferr missing data, identify unusual data, and possibly discover new information and relationships.

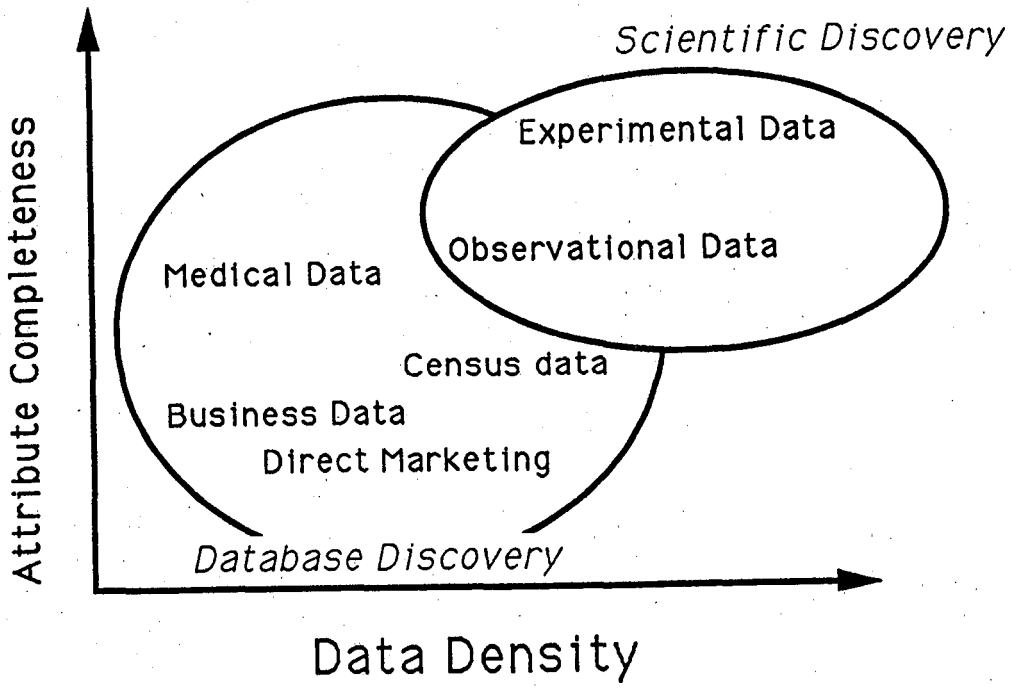
Knowledge Discovery Workbench

Gregory Piatetsky-Shapiro
Christopher J. Matheus

GTE Laboratories, M/S 45
40 Sylvan Road
Waltham, MA 02254

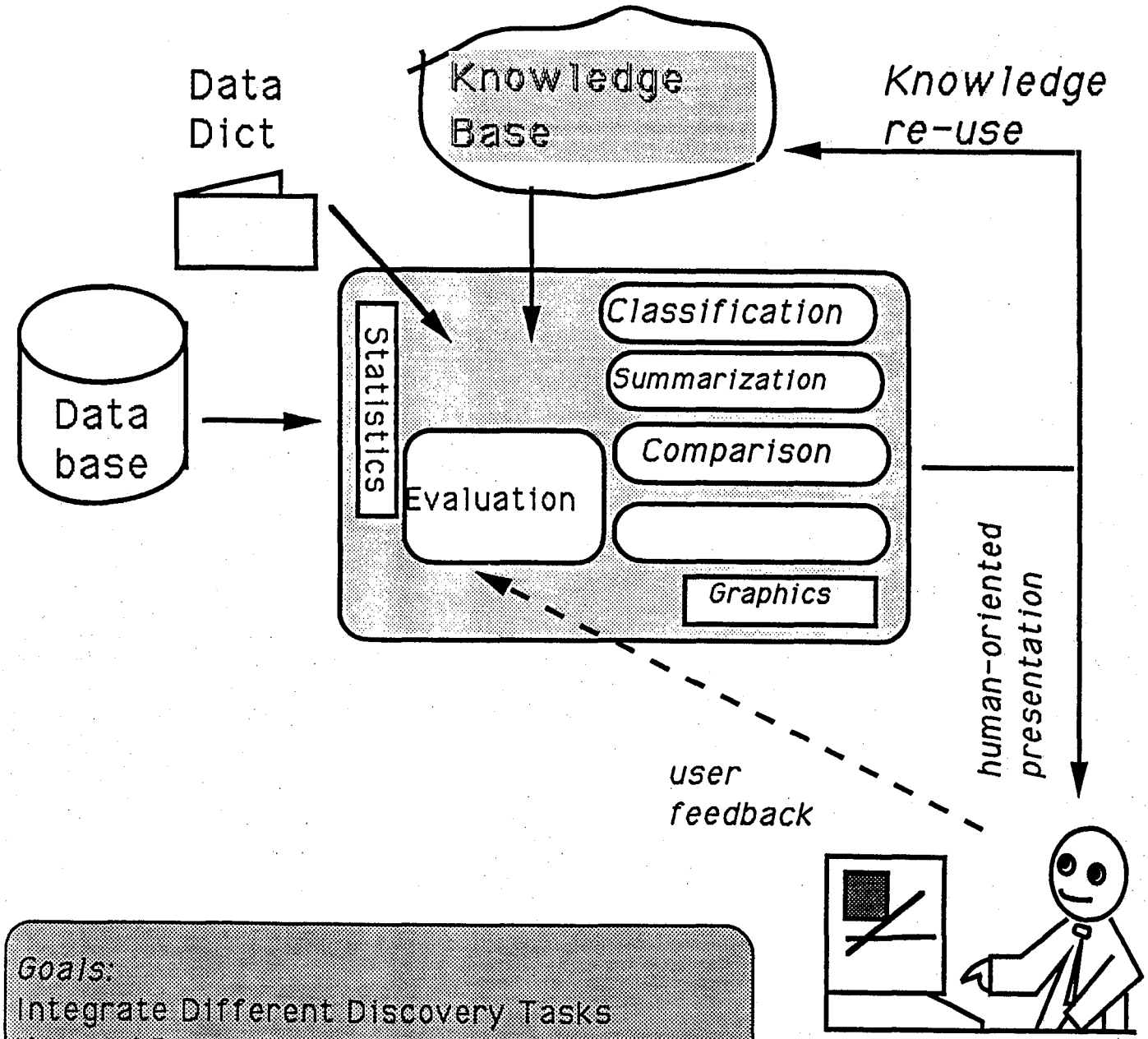
E-mail: gps0@gte.com

Scientific Discovery vs. Database Discovery

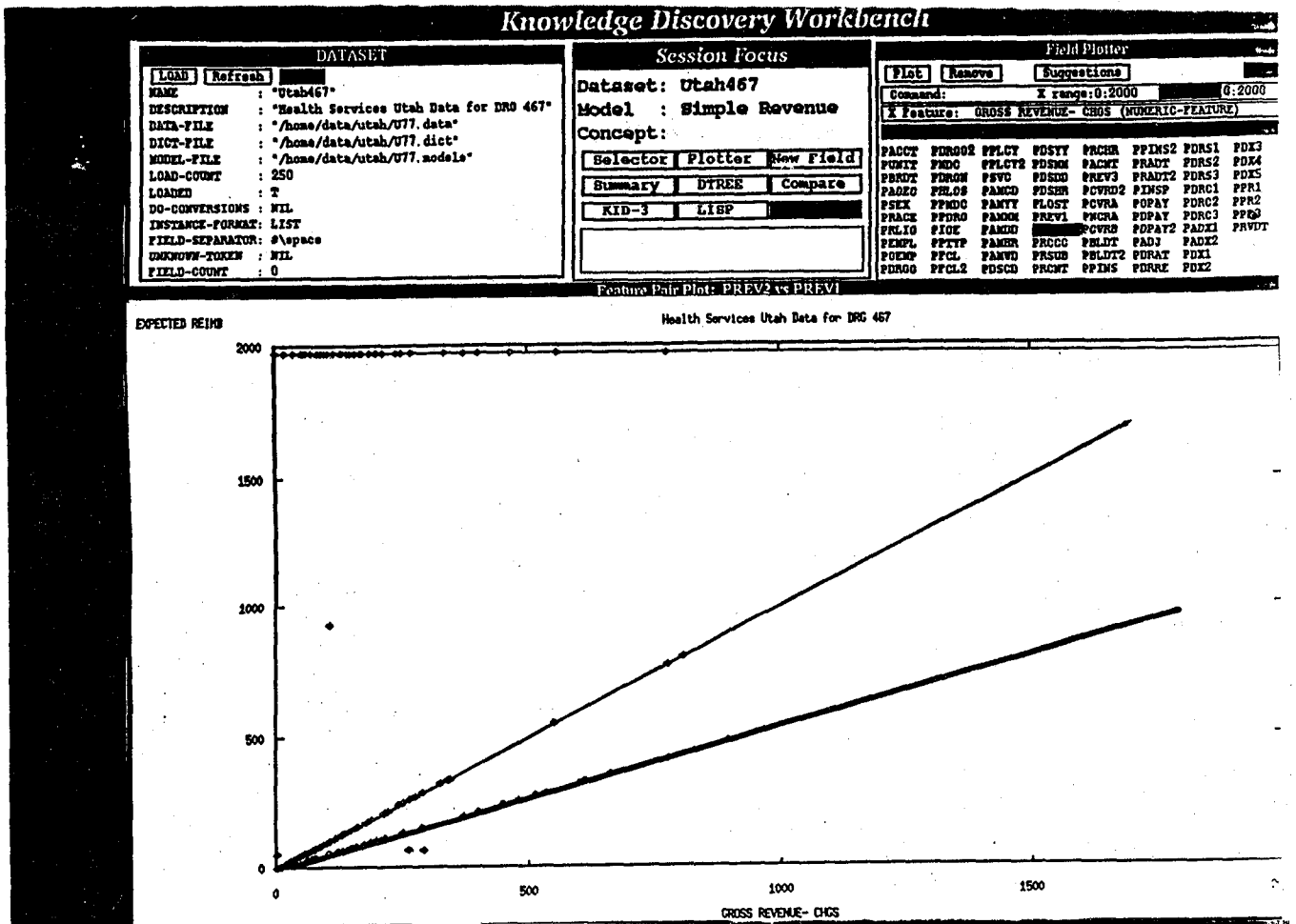


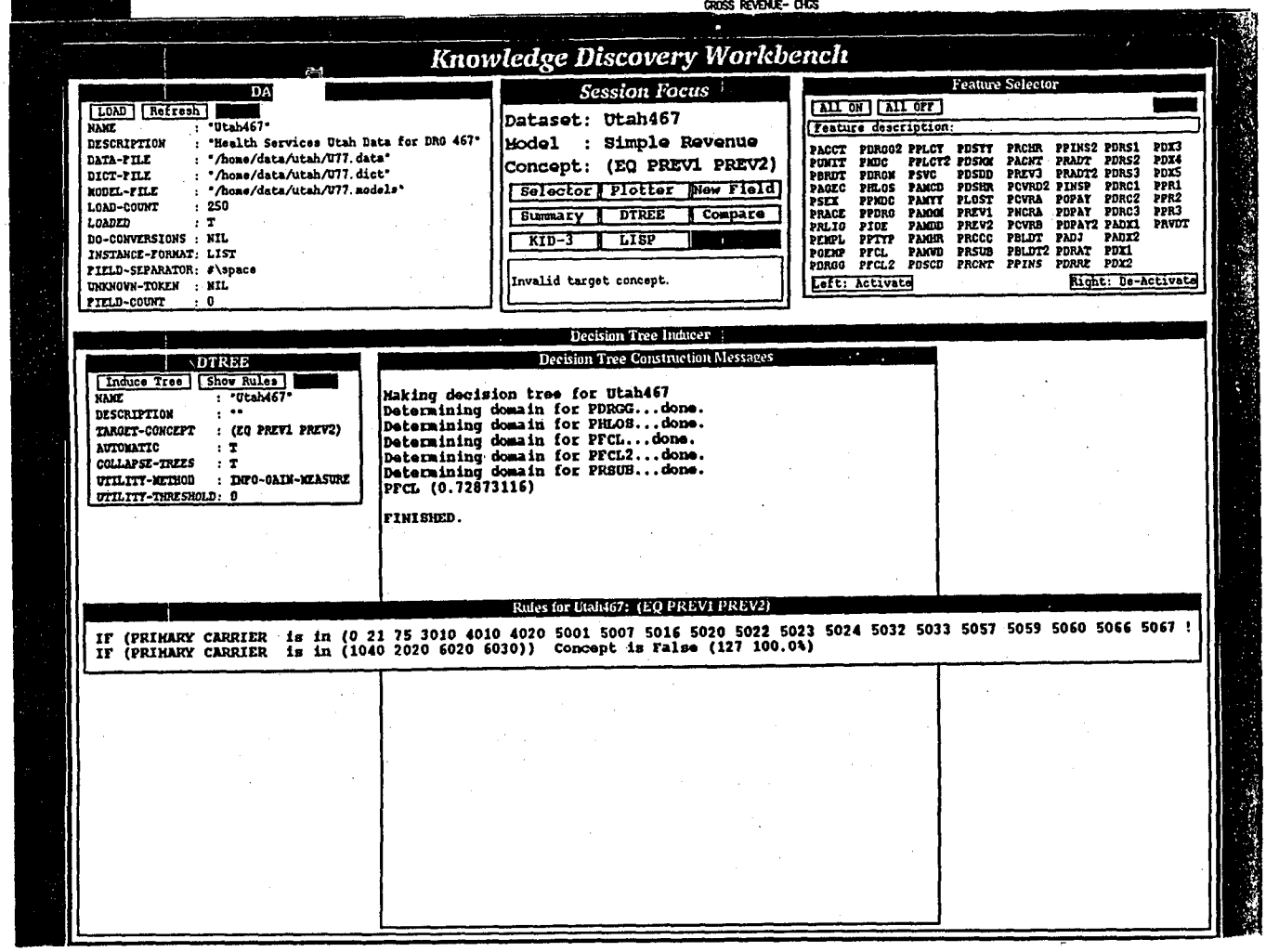
Scientific Discovery	*Database Discovery*
Usually Quantitative, Precise	Frequently Qualitative, Imprecise
Control over data collection. Relevant Attributes. High instance density.	Data collected for a different purpose. Incomplete attributes. Low instance density.
Objective, noncontroversial data.	if data concerns people discovery may be unethical or illegal

KDW Architecture



Goals:
Integrate Different Discovery Tasks
Use and Reuse Domain Knowledge
Use sampling and statistics for large DB
Human-oriented, visual presentation
Read existing DB and data dict


Knowledge Discovery Workbench
DATASET
LOAD
Refresh
NAME : "Utah467"
DESCRIPTION : "Health Services Utah Data for DRG 467"
DATA-FILE : "/home/data/utah/U77.data"
DICT-FILE : "/home/data/utah/U77.dict"
MODEL-FILE : "/home/data/utah/U77.models"
LOAD-COUNT : 250
LOADED : T
DO-CONVERSIONS : NIL
INSTANCE-FORMAT: LIST
FIELD-SEPARATOR: #\space
UNKNOWN-TOKEN : NIL
FIELD-COUNT : 0
Session Focus
Dataset: Utah467
Model : Simple Revenue
Concept:
Selector
Plotter
New Field
Summary
DTREE
Compare
KID-3
LISP
Field Plotter
Plot
Remove
Suggestions
Command:
X range:0:2000
0:2000
X Feature: CROSS REVENUE- CHGS (NUMERIC-FEATURE)
PACCT PDRG02 PPLCY PDSTY PRCHR PPINS2 PDRS1 PDX3
PUNIT PMDC PPLCT2 PDSMM PACMT PRADT PDRS2 PDX4
PBRDT PDRGN PSVC PDSDD PREV3 PRADT2 PDRS3 PDX5
PAGEC PHLOS PAMCD PDSHR PCVRD2 PINSP PDRC1 PPR1
PSEX PPMDC PAMTY PLOST PCVRA PDPAY PDRC2 PPR2
PRACE PPDRG PAMDM PREV1 PNCRA PDPAY PDRC3 PPR3
PRLIG PIDE PAMDD PREV2 PCVRB PDPAT2 PADX1 PRVDT
PEMPL PPTYP PAMHR PRCCC PBLDT PADJ PADX2
POEMP PFCL PAMWD PRSUB PBLDT2 PDRAT PDX1
PDRGG PFCL2 PDSCD PRCNT PPINS PDRRE PDX2
Feature Pair Plot: PREV2 vs PREV1
EXPECTED REIMB
Health Services Utah Data for DRG 467
2000
1500
1000
500
0
0
500
1000
1500
CROSS REVENUE- CHGS


Knowledge Discovery Workbench
DA
LOAD
Refresh
NAME : "Utah467"
DESCRIPTION : "Health Services Utah Data for DRG 467"
DATA-FILE : "/home/data/utah/U77.data"
DICT-FILE : "/home/data/utah/U77.dict"
MODEL-FILE : "/home/data/utah/U77.models"
LOAD-COUNT : 250
LOADED : T
DO-CONVERSIONS : NIL
INSTANCE-FORMAT: LIST
FIELD-SEPARATOR: #\space
UNKNOWN-TOKEN : NIL
FIELD-COUNT : 0
Session Focus
Dataset: Utah467
Model : Simple Revenue
Concept: (EQ PREV1 PREV2)
Selector
Plotter
New Field
Summary
DTREE
Compare
KID-3
LISP
Invalid target concept.
Feature Selector
ALL ON
ALL OFF
Feature description:
PACCT PDRG02 PPLCY PDSTY PRCHR PPINS2 PDRS1 PDX3
PUNIT PMDC PPLCT2 PDSMM PACMT PRADT PDRS2 PDX4
PBRDT PDRGN PSVC PDSDD PREV3 PRADT2 PDRS3 PDX5
PAGEC PHLOS PAMCD PDSHR PCVRD2 PINSP PDRC1 PPR1
PSEX PPMDC PAMTY PLOST PCVRA PDPAY PDRC2 PPR2
PRACE PPDRG PAMDM PREV1 PNCRA PDPAY PDRC3 PPR3
PRLIG PIDE PAMDD PREV2 PCVRB PDPAT2 PADX1 PRVDT
PEMPL PPTYP PAMHR PRCCC PBLDT PADJ PADX2
POEMP PFCL PAMWD PRSUB PBLDT2 PDRAT PDX1
PDRGG PFCL2 PDSCD PRCNT PPINS PDRRE PDX2
Left: Activate
Right: De-Activate
Decision Tree Inducer
DTREE
Induce Tree
Show Rules
NAME : "Utah467"
DESCRIPTION : ""
TARGET-CONCEPT : (EQ PREV1 PREV2)
AUTOMATIC : T
COLLAPSE-TREES : T
UTILITY-METHOD : INFO-GAIN-MEASURE
UTILITY-THRESHOLD: 0
Decision Tree Construction Messages
Making decision tree for Utah467
Determining domain for PDRGG...done.
Determining domain for PHLOS...done.
Determining domain for PFCL...done.
Determining domain for PFCL2...done.
Determining domain for PRSUB...done.
PFCL (0.72873116)
FINISHED.
Rules for Utah467: (EQ PREV1 PREV2)
IF (PRIMARY CARRIER is in (0 21 75 3010 4010 4020 5001 5007 5016 5020 5022 5023 5024 5032 5033 5057 5059 5060 5066 5067 !
IF (PRIMARY CARRIER is in (1040 2020 6020 6030)) Concept is False (127 100.0%)

Summarizing
Expected Reimbursement = 19.74

Found 181 instances with this condition. These instances also have these features:

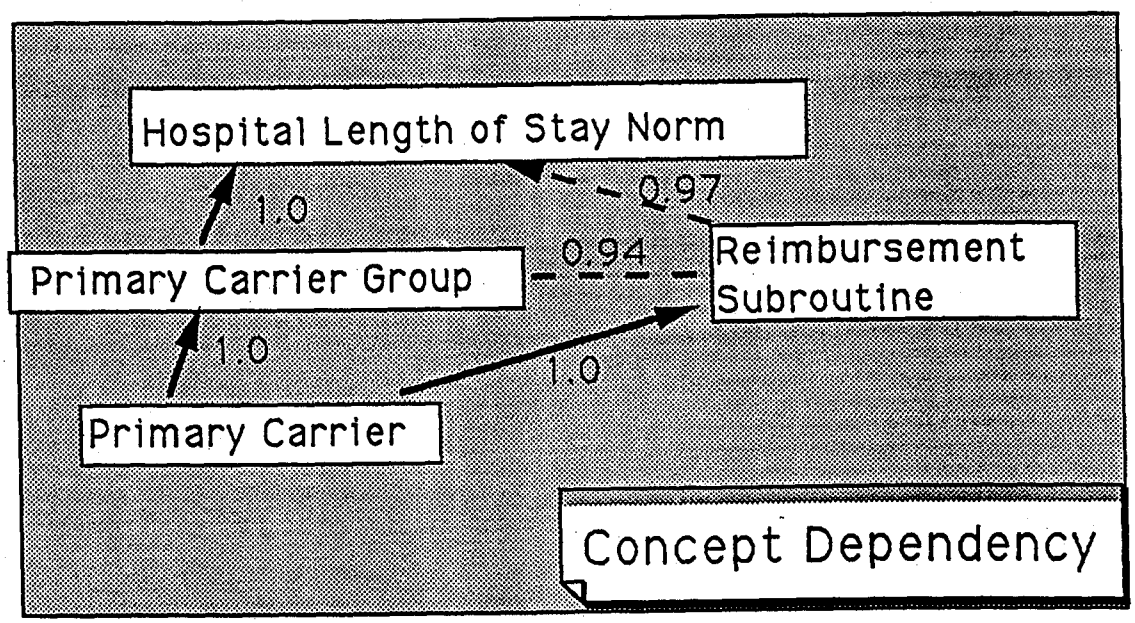
Primary Carrier Group= 22 (count 181)

Hospital Length of Stay Norm = 4 (181)

Primary Carrier = 2020 (181)

Reimbursement Subroutine = DGRS556 (183)

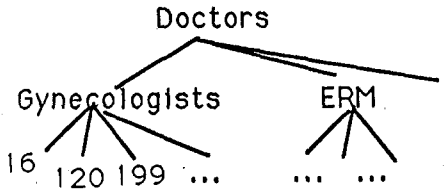
. . .



After pruning:

Primary Carrier = 2020 (181)

Comparative Discovery




```
*** analyzing PHYSN - ATTENDING = 16 (7 records)
has PATIENT AGE (PAGEC) = 51.57, *significantly* above (1.04 std)
the overall average of 34.51 (overall std=16.48).

has EXPECTED REIMB (PREV2) = 88.43, somewhat below (-0.97 std)
the overall average of 984.21 (overall std=921.88).

has OTHER PAYMENTS (POPAY) = -56.14, somewhat below (-0.92 std)
the overall average of -16.4 (overall std=43.39).

*** analyzing PHYSN - ATTENDING = 120 (20 records)

*** analyzing PHYSN - ATTENDING = 199 (26 records)
```

Use of Domain Knowledge

	Example	Use	Source
Data Dictionary	Field type, size, possible values	Optimize search Type-specific knowledge	Data Dict.
Field Interest	AcctNo ignore WeekDay low Insurance high	focus search evaluation	Field Stats
Field Relationships	Financial Expected Reimburs. Charges ...	focus search operator applicability	
Field Value Taxonomy	Customer-Type Large-Organization Business Residential FED GTE IXC E Employee Regular A B D H G	Bias to better generalization. Human-oriented presentation	
Field Dependencies	PFCL 0.95 PHLOS REV1 ≤ REV2 *(usually?)* YTD = QT1 +QT2 + QT3 + QT4	Prune Search Flag Exceptions. Don't rediscover what is known	
Value Dependencies (Rules)	DType = GYN *always* P-sex = F DType = GYN 15 < P-age < 65	MDC 22 DRG 457 458-60 472 DX 94890-9 9490-5	KID3

KDW main features

- Integration of different discovery tasks
- Use of domain knowledge and re-use of discovered knowledge
- Interactive, human-oriented tool
- Use on large business databases

Future Directions

- Additional discovery tasks - clustering, forecasting, ...
- Integrate Database Operations with Discovery Methods (Database Discovery Algebra ?)
- Data and Knowledge Visualization
- ???

Knowledge Discovery in Databases – An Overview

William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus
GTE Laboratories Incorporated

Computers have promised us a fountain of wisdom
but delivered a flood of data
A frustrated MIS executive

This chapter presents an overview of the state of the art in research on Knowledge Discovery in Databases. We analyze what is Knowledge Discovery, *and define it as the non-trivial extraction of implicit, previously unknown, and potentially useful information from data. We then compare and contrast database, machine learning and other approaches to discovery in data. We present a framework for knowledge discovery, and examine problems of dealing with large, noisy databases, the use of domain knowledge, the role of the user in the discovery process, discovery methods, and the form and uses of discovered knowledge.*

We also discuss application issues, including the variety of existing applications, and propriety of discovery in social databases. We present criteria for selecting an application in a corporate environment. In conclusion, we argue that discovery in databases is both feasible and practical and outline directions for future research, which include better use of domain knowledge, efficient and incremental algorithms, interactive systems, and integration on multiple levels.

It has been estimated that the amount of information in the world doubles every 20 months. The size and number of databases probably increases even faster. In 1989, the total number of databases in the world was estimated at five million, although most of them are small dbaseIII^{TM} databases. The automation of business activities produces an ever-increasing stream of data, because even simple transactions, such as a telephone call, the use of a credit card, or a medical test, are typically recorded in a computer.

Scientific and government databases are also rapidly growing. The National Aeronautics and Space Administration already has much more data than it can analyze. Earth observation satellites, planned for 1990s, are expected to generate one terabyte (10^{15} bytes) of data every day – more than all previous missions combined. At a rate of one picture each second, it would take a person several years (working nights and weekends) just to look at the pictures generated in one day. In biology, the federally funded Human Genome project will store thousands

of bytes for each of the several billion genetic bases. Closer to everyday lives, the 1990 U.S. census data of a million million bytes encode patterns that in hidden ways describe the lifestyles and subcultures of today's United States.

What are we supposed to do with this flood of raw data? Clearly, little of it will ever be seen by human eyes. If it will be understood at all, it will have to be analyzed by computers. Although simple statistical techniques for data analysis were developed long ago, advanced techniques for intelligent data analysis are not yet mature. As a result, there is a growing gap between data generation and data understanding. At the same time, there is a growing realization and expectation that data, intelligently analyzed and presented, will be a valuable resource to be used for a competitive advantage.

The computer science community is responding to both the scientific and practical challenges presented by the need to find the knowledge adrift in the flood of data. In assessing the potential of AI technologies, Donald Michie (1990), a leading European expert on machine learning, predicted that "the next area that is going to explode is the use of machine learning tools as a component of large-scale data analysis". A recent National Science Foundation workshop on the future of database research ranked "data mining" among the most promising research topics for the 1990s (Silberschatz, Stonebraker and Ullman 1990). Some research methods are already well enough developed to have been made part of commercially available software. Several expert system shells use variations of ID3 for inducing rules from examples. Other systems use inductive, neural net, or genetic learning approaches to discover patterns in personal computer databases.

Many forward-looking companies are using these and other tools to analyze their databases for interesting and useful patterns. American Airlines searches its frequent flyer database to find its better customers, targeting them for specific marketing promotions. *Farm Journal* analyzes its subscriber database and uses advanced printing technology to custom-build hundreds of editions tailored to particular groups. Several banks, using patterns discovered in loan and credit histories, have derived better loan approval and bankruptcy prediction methods. General Motors is using a database of automobile trouble reports to derive diagnostic expert systems for various models. Packaged-goods manufacturers are searching the supermarket scanner data to measure the effects of their promotions and to look for shopping patterns.

A combination of business and research interests has produced increasing demands for, as well as increased activity to provide tools and techniques for discovery in databases. This book is the first to bring together leading-edge research from around the world on this topic. It spans many different approaches to discov-

Models and the Bayesian Classification Of Protein Structural Elements

David States and Lawrence Hunter
National Center for Biotechnology Information
and the Lister Hill Center
National Library of Medicine, Bethesda, MD 20894

Abstract

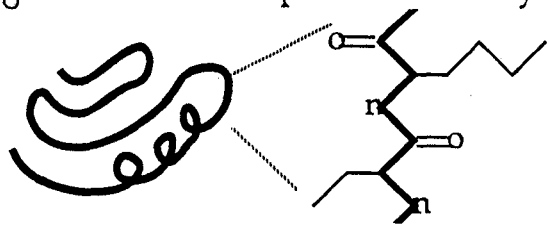
Motifs are present in protein structure, but a precise delineation of these elements has proven elusive, and estimates of the number and individual characteristics of such patterns vary widely in the literature. To place the analysis of protein structure on a rational basis, the technique of Bayesian classification has been applied to the categorization of protein structural elements. The use of statistical mechanics as model for the relationship of structure and probabilities in chemical systems is presented. A data set of 9556 segments of peptide backbone structure, each five amino acids in length, derived from 53 high resolution structures from the Brookhaven PDB, was used for analysis. Classifications based on independent cartesian coordinates of the backbone atoms demonstrated the presence of 27 classes varying in both mean coordinates and the extent of variation about the mean coordinate. These classes correlate well with classically defined elements of secondary structure, but subdivisions based on heterogeneity in coordinate variance are apparent and, in some cases, as characteristic as are coordinate means. Strong biases in amino acid sequence are seen at particular sites for several classes, but the predictive power of these correlations is weak. The type II beta turn is recognized as a class and shown not to require a glycine residue at position 3. The formalism of Bayesian classification can also be applied to the analysis of substates in molecular dynamics trajectories. Using butane as an example, the importance of considering cross correlation terms in the analysis of trajectyories is discussed. When arbitrary co-variance terms are considered in the class model of protein structure, as few as eight classes are required to describe the data.

Bayesian Classification of Macromolecular Structural Motifs

Protein structure determines activity

Proteins are chains of amino acids

Folding of the chain is required for activity



600 protein known structures

Proposed classifications of protein structure range frome 3 to 250 classes

Bayesian classification

A rational basis for determining the number and types of structural motifs

Data set: 9,656 fragments of protein structure derived from high resolution protein structures

'Autoclass III identified 27 classes of structural motifs

The Bayesian classes vary in both mean coordinates and variation about the mean

Correlations are observed between the amino acid sequence and the structural classes

Applications of Bayesian Classification in Chemistry and Biophysics

Data Sets

Three-dimensional structure determination
- 80,000 small molecule structures
- 600 macromolecular structures

Molecular modeling and dynamical simulations

Molecular sequences databases
- 26,000 protein sequences
- 39,000 nucleic acid sequences
- human genome initiative -> $6x10^9$ bases of nucleic acid sequence

Applications for Classification

Identification of structural motifs

Reduction of dynamical systems to a description based on sub-states and transitions

Identification of sequence motifs and families

Structure function relationships

Statistical mechanics

A desciprition based on the average properties of a system with infinitely many particles.

Relating energy to probabilities

Boltzmann's constant relates potential energies to probabilities

$$\frac{P(a)}{P(b)} = e^{\frac{-\Delta E_{ba}}{kT}}$$

Potential energies can be calculated

ab initio methods: based on quantum mechanics directly

Empirical methods

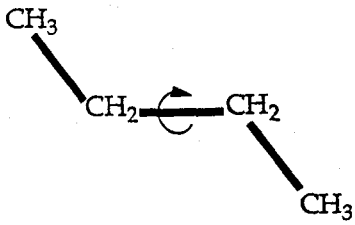
The potential energy surfaces determines the dynamical properties of the system

Normal modes

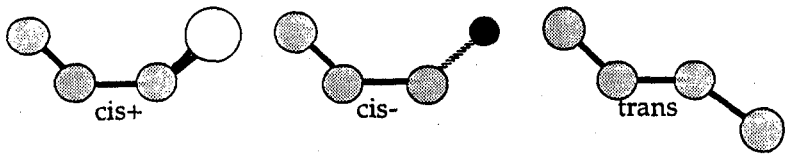
Harmonic energy well -> Gaussian spatial distribution

Applying the Autoclass Model to Butane

Butane



The potential energy surface for butane has three minima separated by moderate barriers



Autoclass III analysis

Structural snap shots from molecular dynamics analyzed with Autoclass III -> 6 classes

None of the classes correspond to minimum energy configurations of the system

Limitations of the Autoclass model

Independence assumption: atomic positions in chemical systems are highly correlated

- bonds and bond angle restrictions
- packing forces.

Bayesian Classification of Correlated Data

Correlated data

Normal distributions in N dimensions

Matrix of co-variance terms

$$P(r) = \sqrt{\frac{det(V)}{(2\pi)^N}} e^{-\frac{(r-\bar{r})V(r-\bar{r})}{2^N}}$$

Classes may differ in:

- Mean coordinates
- Coordinate variation
- Coordinate correlations

Costs of the extension

N mean and N(N+1)/2 co-variance terms per class

More free values to fit -> more data required to adequately determine the problem.

Order N^4 or worse computational complexity

Optimization pitfalls

zero or negative eigenvalues

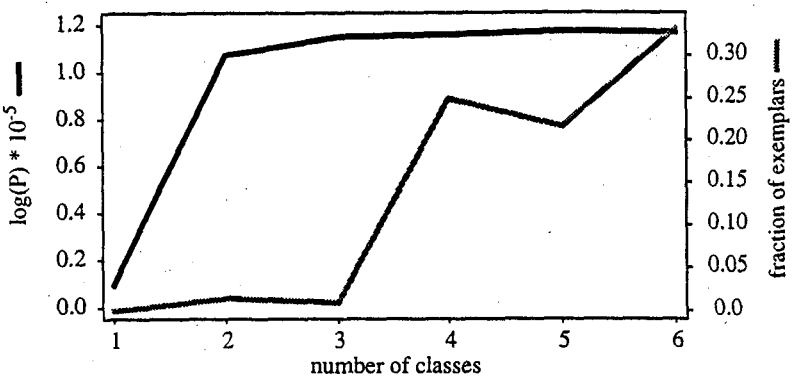
collapse to zero volume with insufficient data

Is Data Classifiable?

Data set

10,000 snapshots from a trajectory of butane at 600K.

Results

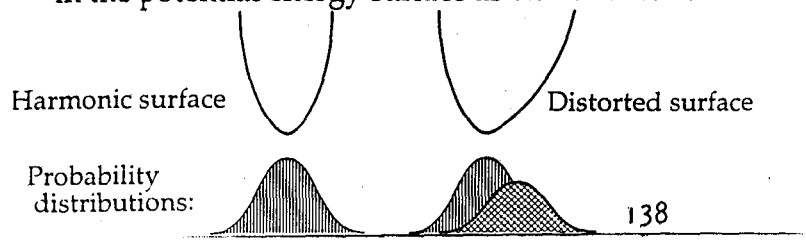


Interpretation

Three class case corresponds to physical intution

Appropriate means and coordinate correlations.

Additional classes appear to represent anharmonicities in the potential energy surface as sums of Gaussians.



Derived co-variance terms

Second derivatives of the potential energy surface -> normal modes -> probability distribution

Avoid model consistence issues

Informative prior probabilities

Incorporating other chemical knowledge
- bond lengths and geometries
- packing constraints
- force constants

Boltzmann's law -> relative class probabilities

Well bredth -> entropy

Implications for computation

Fewer free values to fit reduces data requirements

Fewer free values to fit reduces computational costs

Additional constraints -> improved behavior in optimization

Summary and Conclusions

Bayesian classification methods are useful in chemistry and biophysics.

Chemical physics provides a rigorous model for analysis.

Models are important:

The wrong model -> misleading results.

A good model extends the range and power of classification.

Even approximate models may be a big win.

Aspects of Astronomical Research Involving Large Data Bases

Nick Weir and Stan Djorgovski
California Institute of Technology

The enormous size and information content of many astronomical data sets, particularly those derived from space-based missions, render traditional means of data analysis unfit for the task. Researchers have begun to more vigorously investigate automated, multivariate means of analyzing and classifying large-scale data bases, drawing heavily upon recent efforts in statistics and pattern recognition. Not only are such methods the only ones equipped to handle unfathomably large amounts of data efficiently, but, as astronomers are learning, they are incomparably superior to traditional methods in their ability to extract important information from the data.

We review recent work in this area of astronomy, providing examples of both supervised and unsupervised pattern analysis and discovery. Investigations in supervised learning include efforts to design efficient star/galaxy discriminators, specialized stellar and quasar identifiers, and optimal filter combinations for predicting redshifts of galaxies from their colors, all of which are relevant to the analysis of current large-scale imaging surveys. Systems for sorting other types of astronomical data into predetermined classes are discussed as well. Some of the most of exciting research, however, involves the application of unsupervised techniques to the areas of spectral and image classification, and the multivariate analysis of astronomical catalogs. Enthusiasm for this type of research is growing as astronomers discover it contributes to physical understanding in previously unexplored ways.

ASPECTS OF ASTRONOMICAL RESEARCH INVOLVING VERY LARGE DATA BASES

Nicholas Weir
S. Djorgovski

Department of Astronomy
California Institute of Technology

Outline:

- Nature of the Data
- Current Approaches
- Our project: Digitized POSS II
- Where we are headed

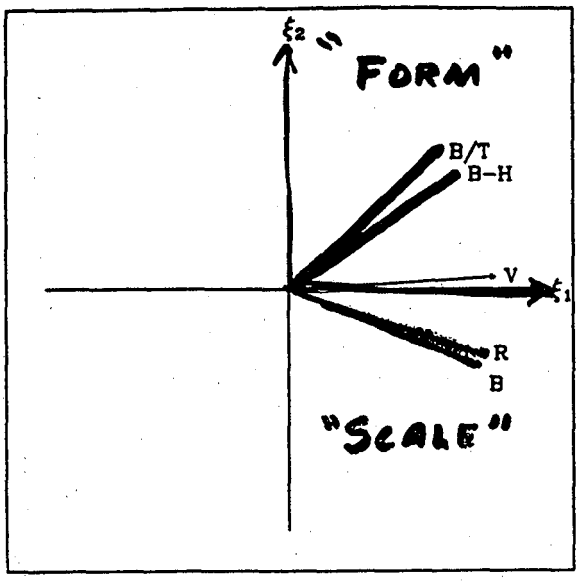
Nature of the Data

- o Primarily imaging and spectroscopy initially
- o Reduced to catalog form
- o Voluminous
  - Space-based missions
  - Ground surveys
  - General observations (mostly discarded)

Current Approaches

- o Traditional multivariate statistical techniques
  - Factor Analysis (PCA)
    Unsupervised classification and correlative studies
    e.g., Elliptical and spiral galaxies
    IRAS sources
  - Cluster Analysis
    Unsupervised classification
    Supervised classification
    e.g., MK spectral classification
    Quasar searches
  - Discriminant Analysis
    Supervised classification
    e.g., Star/Galaxy seperation
    IUE spectra

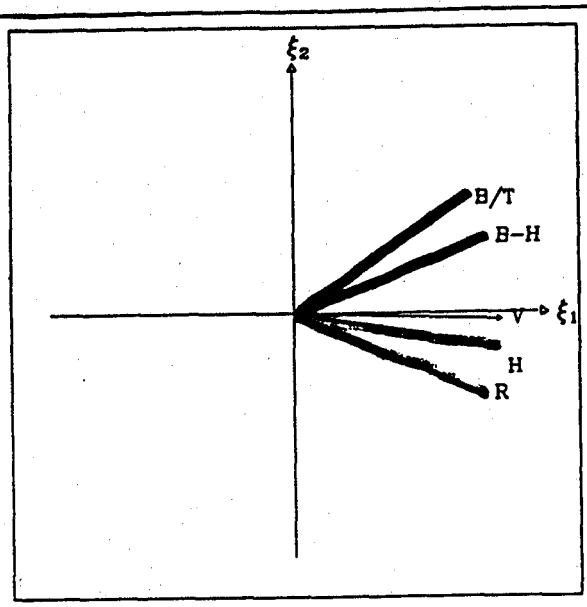
Principal Component Analysis of Spiral Gal's



Spirals (Whitmore)
51 galaxies

Var_1 = 54.5 %
Var_2 = 31.3 %

2 dominant dimensions



Spirals (Whitmore)
51 galaxies

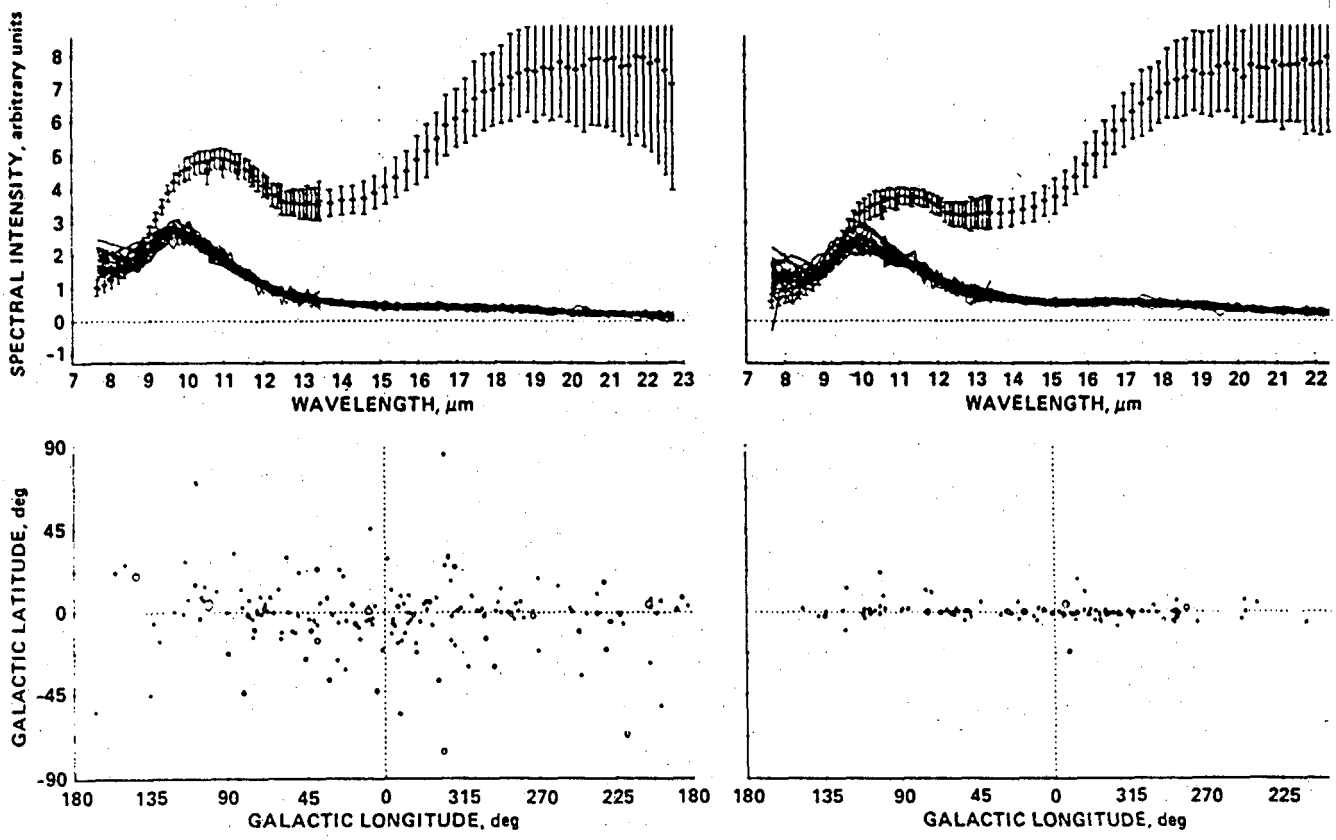
Var_1 = 61.2 %
Var_2 = 25.1 %

Current Approaches

o Non-traditional methods

- AUTOCLASS
  Unsupervised classification
  e.g., IRAS spectra

Goebel et al. (1989)



- Artificial Neural Networks
  Supervised classification
  e.g., z(color) predictor
  Phasing mirror arrays

Our Project

Digitized 2nd Palomar Observatory Sky Survey

- Complete Northern Sky survey in three bands
- > 3 Terabytes of pixels result from the scans
- > 10^8 stars and 10^7 galaxies will be detectable

Task:

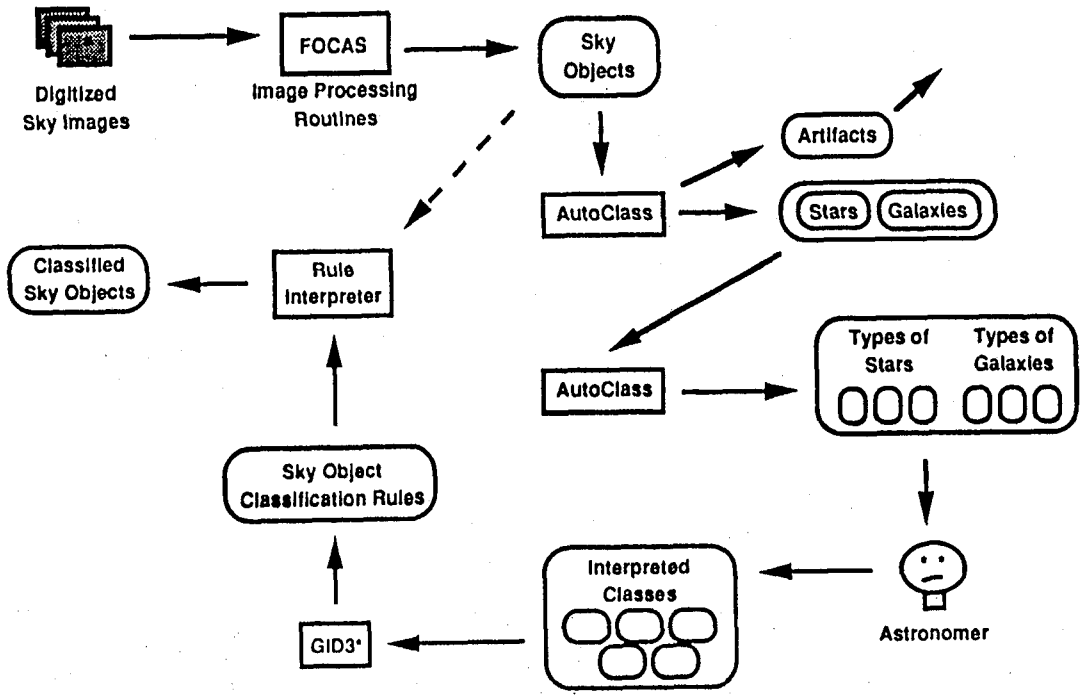
- Measure, classify, and analyze all objects detected in the scans
- Do so in a fully objective and uniform manner

Collaborative effort between JPL and Caltech

- Usama Fayad
  Richard Doyle
  Artificial Intelligence Group
  Jet Propulsion Laboratory
- S. Djorgovski
  Nicholas Weir
  Department of Astronomy
  Caltech

JPL

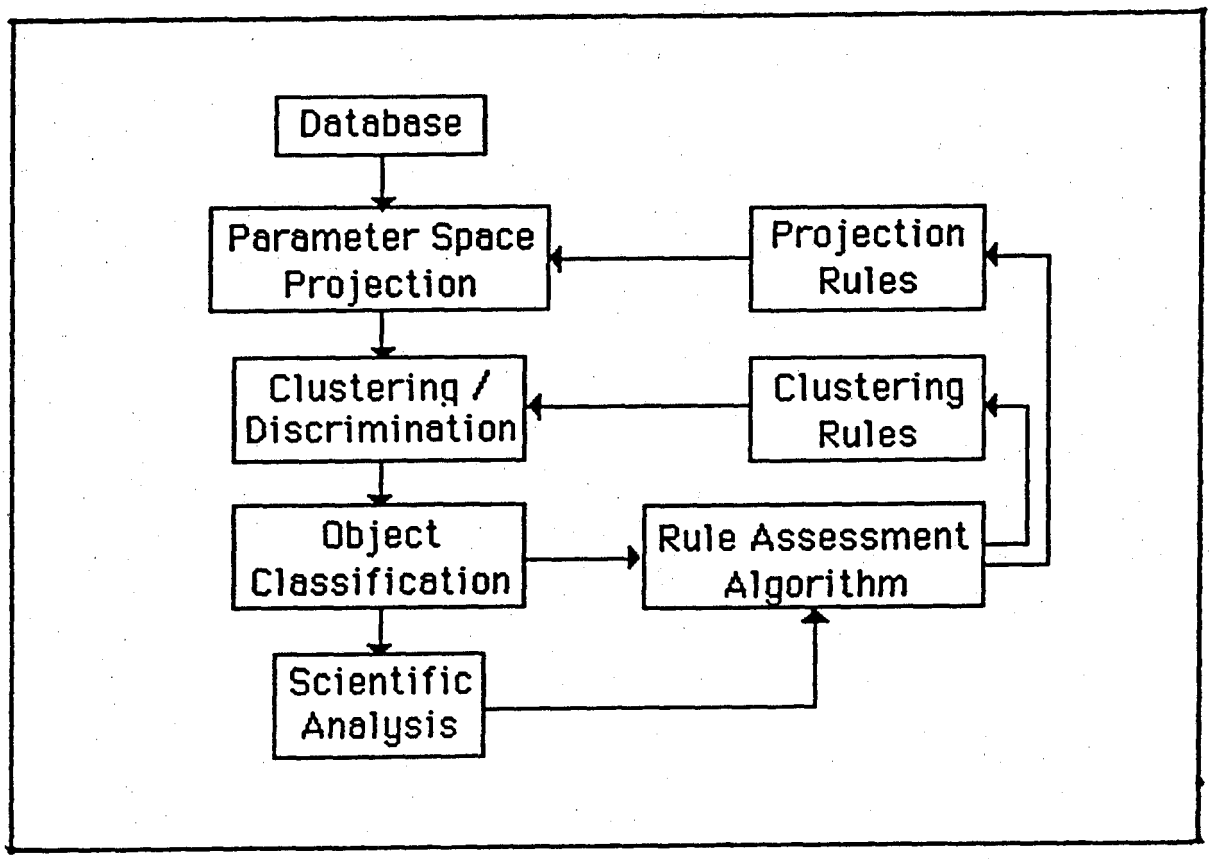
APPROACH



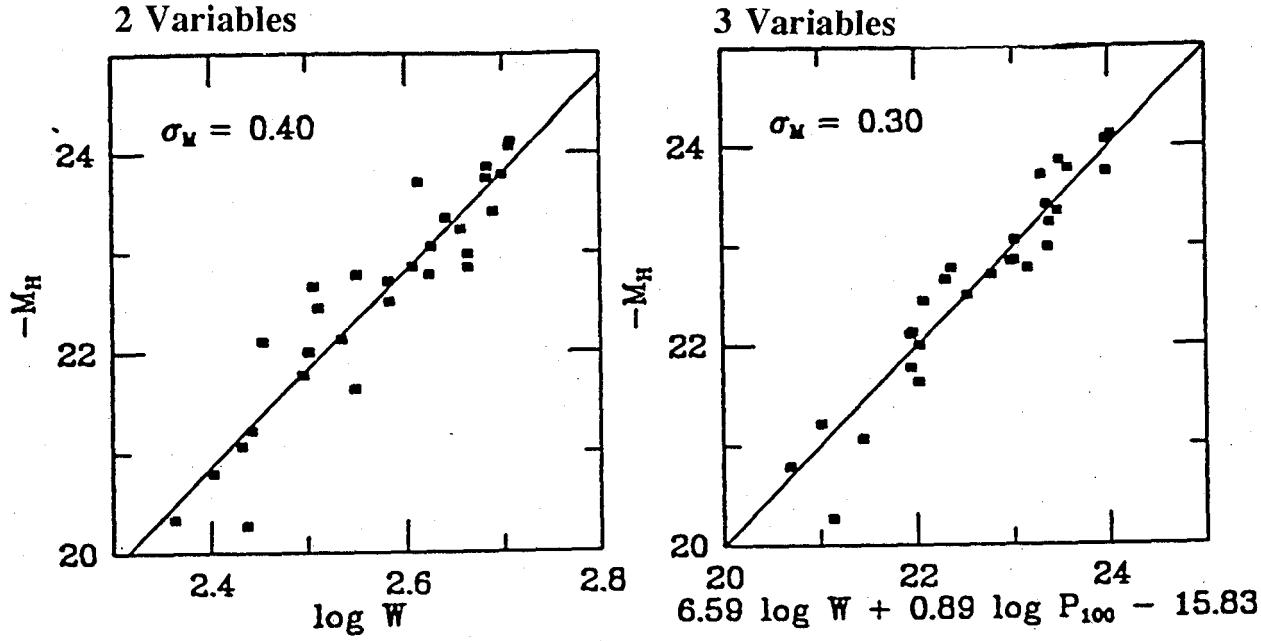
SAA

Where we are headed...

- Generalized multivariate analysis of large, heterogeneous data sets
  - Integration
  - Projection
  - Classification
  - Analysis

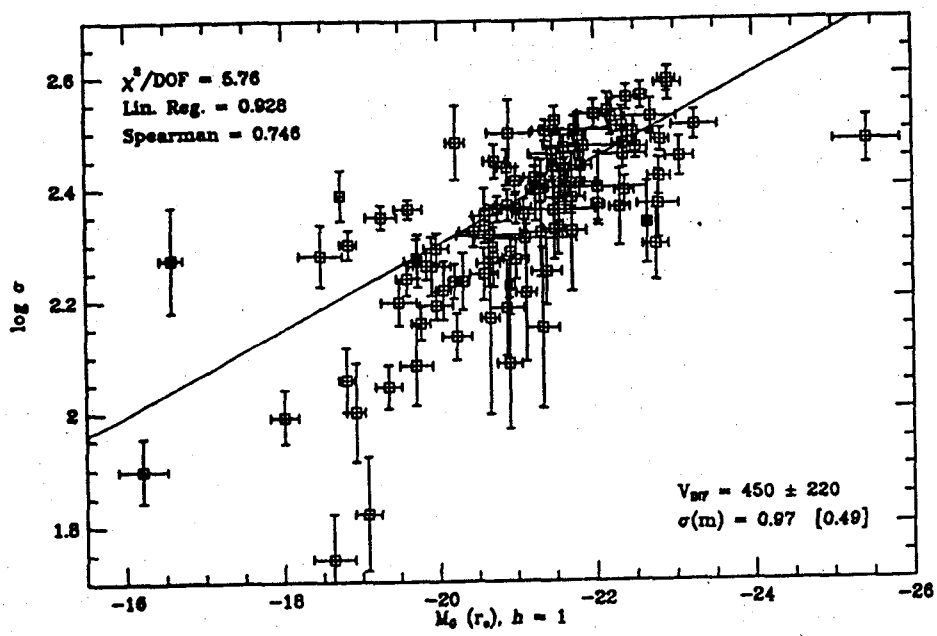


- Anticipated result
  - Greater physical understanding
    e.g., IR-IRAS Tully-Fisher relation
    "Fundamental Plane" of elliptical galaxies



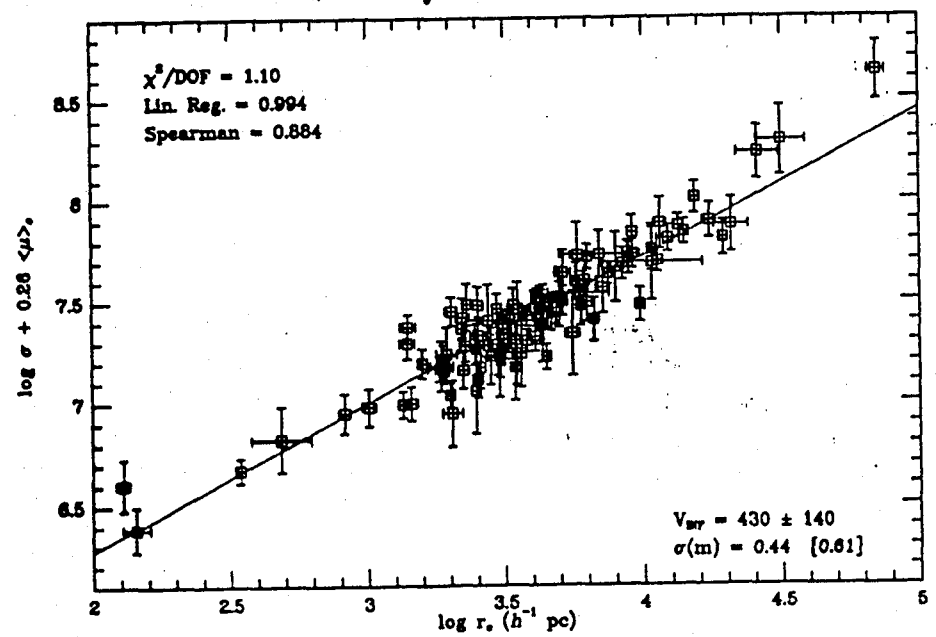
Left: the H-band Tully-Fisher relation for the late-type spirals ($T \geq 5$) in the Aaronson LSC Best sample. Right: the IRAS-corrected Tully-Fisher relation. The r.m.s. scatter in absolute magnitudes is indicated. Much of the residual scatter is probably due to the distance errors (we used a very simple linear Virgo infall model).

FJ RELATION



DJORGOVSKI & DAVIS SAMPLE

R-σ-μ RELATION



DD SAMPLE

Conclusion

- Astronomers are investigating more sophisticated tools out of necessity: the data are too vast.

- They are (slowly) discovering the general desirability of these methods in the process.

REFERENCES

Contributed by Nick Weir.

Books of interest:

Di Gesu, V. *et al.* (eds.) 1984, *Data Analysis in Astronomy.* New York: Plenum Press.

Di Gesu, V. *et al.* (eds.) 1986, *Data Analysis in Astronomy II.* New York: Plenum Press.

Jascheck, C., and Murtagh, F. (eds.) 1990, *Errors, Bias and Uncertainties in Astronomy.* Cambridge: Cambridge University Press.

Murtagh, F., and Heck, A. 1987, *Multivariate Data Analysis.* Dordrecht: Reidel.

Murtagh, F., and Heck, A. (eds.) 1988, *Astronomy from Large Databases: Scientific Objectives and Methodological Approaches.* Garching bei Munchen: European Southern Observatory.

Murtagh, F., and Heck, A. (eds.) 1989, *Knowledge Based Systems in Astronomy.* Berlin: Springer-Verlag.

Statistical Methods in Astronomy, 1983. European Space Agency Special Publication 201.

Applications of Factor or Principal Component Analysis:

Brosche, P. 1973, *Astronomy and Astrophysics* **23**, 259.

Djorgovski, S., and Davis, M. 1987, *Astrophysical Journal* **313**, 59.

Djorgovski, S., and de Carvalho, R. 1990, in Fabbiano, G. *et al.* (eds.), *Windows on Galaxies*, Erice Astrophysics Workshop, in press. Dordrecht: Kluwer.

Whitmore, B. C. 1984, *Astrophysical Journal* **278**, 61.

Applications of Unsupervised Classification and Clustering:

Bianchi, R., Coradini, A., and Fulchignoni, M. 1980, *The Moon and the Planets* **22**, 293.

Bianchi, R., *et al.* 1980, *The Moon and the Planets* **22**, 305.

Butchins, S., 1982, *Astronomy and Astrophysics* **109**, 360.

Carusi, A., and Massaro, E. 1978, *Astronomy and Astrophysics Suppl.* **34**, 81.

Cheeseman, P., *et al.* 1988, in *Proc. Fifth Machine Learning Workshop*, p.54. Morgan Kaufman.

Denning, P. 1989, *American Scientist* **77**, 216.

Giovannelli, F., Coradini, A., Lasota, J., and Polimene, M. 1981, *Astronomy and Astrophysics* **95**, 138.

Goebel, J. *et al.* 1989, *Astronomy and Astrophysics* **222**, L5.

Mennessier, M. 1985, *Astronomy and Astrophysics* **144**, 463.

Sebok, W., 1979, *Astronomical Journal* **84**, 1526.

Valdez, F. 1982, *Proc. SPIE* **331**, 465.

Applications of Discriminant Analysis:

Heck, A., Egret, D., Jaschek, M., and Jaschek, C. 1984, *IUE Low-Resolution Spectra Reference Atlas - Part 1. Normal Stars.* European Space Agency Special Publication 1052.

Heck, A., Egret, D., Nobelis, P., and Turlot, J. 1986, *Astrophysics and Space Sciences* **120**, 223.

Applications of Supervised Classification:

Jarvis, J., and Tyson, A. 1981, *Astronomical Journal* **86**, 476.

Kurtz, M. 1984, in *The MK Process and Stellar Classification*, ed. R. Garrison, p.136. Toronto: David Dunlap Observatory.

Malagnini, M., Pasian, F., Pucillo, M., and Santin, P. 1985, *Astronomy and Astrophysics* **144**, 49.

Application of Neural Networks:

Angel, J.R.P., Wizinowich, P., Lloyd-Hart, M., and Sandler, D. 1990, *Nature* **348**, 221.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Dates attached	

4. TITLE AND SUBTITLE

Titles/Authors - Attached

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Code FIA - Artificial Intelligence Research Branch
Information Sciences Division

8. PERFORMING ORGANIZATION REPORT NUMBER

Attached

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Nasa/Ames Research Center

Moffett Field, CA. 94035-1000

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Available for Public Distribution

Peter Friedland 5/14/92 BRANCH CHIEF

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Abstracts ATTACHED

14. SUBJECT TERMS			15. NUMBER OF PAGES
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT