

51-63
02-1-1

**INTELLIGENT SYSTEM DEVELOPMENT USING A ROUGH SETS
METHODOLOGY**

**Final Report
NASA/ASEE Summer Faculty Fellowship Program-1993
Johnson Space Center**

Prepared by: Gary T. Anderson, Ph.D.
Academic Rank: Associate Professor
University & Department: University of Arkansas at Little Rock
Department of Applied Science
Little Rock, AR 72204
NASA/JSC
Directorate: Business Information Systems
Division: Information Systems
Office: Information Technology
JSC Colleague: Robert O. Shelton, Ph.D.
Date Submitted: August 9, 1996
Contract Number: NAG 9-867

ABSTRACT


The purpose of this research was to examine the potential of the rough sets technique for developing intelligent models of complex systems from limited information. Rough sets a simple but promising technology to extract easily understood rules from data. The rough set methodology has been shown to perform well when used with a large set of exemplars, but its performance with sparse data sets is less certain. The difficulty is that rules will be developed based on just a few examples, each of which might have a large amount of noise associated with them. The question then becomes, what is the probability of a useful rule being developed from such limited information? One nice feature of rough sets is that in unusual situations, the technique can give an answer of "I don't know". That is, if a case arises that is different from the cases the rough set rules were developed on, the methodology can recognize this and alert human operators of it. It can also be trained to do this when the desired action is unknown because conflicting examples apply to the same set of inputs.

This summer's project was to look at combining rough set theory with statistical theory to develop confidence limits in rules developed by rough sets. Often it is important not to make a certain type of mistake (e.g., false positives or false negatives), so the rules must be biased toward preventing a catastrophic error, rather than giving the most likely course of action. A method to determine the best course of action in the light of such constraints was examined. The resulting technique was tested with files containing electrical power line "signatures" from the space shuttle and with decompression sickness data.


INTELLIGENT SYSTEM DEVELOPMENT USING A ROUGH SETS
METHODOLOGY

Gary T. Anderson
University of Arkansas at Little Rock
BT2
9 August 1996

Robert O. Shelton
Information Technology Office
Information Systems Office
Business Information Systems Directorate



Gary T. Anderson



Robert O. Shelton

INTRODUCTION

As NASA moves forward towards deployment of the space station, a possible permanent manned station on the moon and a manned flight to Mars, the long term reliability and maintenance of life support systems in hostile environments becomes a crucial issue. Intelligent software that can monitor systems and make automated decisions can relieve the human crew of such responsibilities during long space and lunar missions, freeing them to perform other tasks. The complexities of life support and other systems make such software difficult to develop. For example, the software must be able to evaluate several interdependent inputs, with many variations on typical cases. The software should probably also be developed from mission data, which has noise both on system inputs and on system outcomes (i.e., the results of system actions). As a result of these requirements, often very few examples will exist for situations that occur only rarely. It is often in these rare cases where it is critical that the software perform correctly. Rough sets is one technique that makes intelligent monitoring of complex systems less cumbersome.

Since its introduction [Pawlak, 1982], rough sets has proven to be a simple but effective technology to extract rules from data [Slowinski, 1992]. Discussions of rough set methodology are given in [Pawlak, 1988], [Grzymala-Busse, 1988], [Chan, 1991] and [Szladow, 1993]. The rough set technique has been shown to perform well when used with a large set of examples, but its performance with small data sets is less certain. The problem is that rules will be developed based on just a few examples, each of which might have a large amount of noise associated with them. Furthermore, conflicting rules will often be triggered when the system is used on new cases. The question becomes, what is the probability of a useful rule being developed from such limited information? Probability theory is used to determine how much confidence one can have in a given rough set rule based on the number of examples that support that rule.

ROUGH SETS

A core idea in rough sets is that precision is frequently not necessary when looking for patterns in data. For example, a fever is usually enough to indicate the presence of disease, without knowledge of exact body temperature. Rough set rules are generated from a set of examples with known outcomes. Discrete inputs are used as is, while continuous inputs are divided into discrete categories. An input with a temperature range of 96 - 110 °F might be divided into broad ranges described by cold, cool, warm and hot. In this manner, strong patterns that exist in the data are reflected in the model.

The rough sets algorithm is typically used to extract rules from a table of examples. Each example (table row) is called an object, while each piece of information in the example (table column) is called an attribute. The outcome is then called the decision attribute. Ideally, the decision attribute is completely determined by the other attributes in the table, and the outcome is said to be discernible by the inputs. The rough set methodology then searches for a minimal set of input attributes, called reducts, that can

completely describe the decision attribute. Small reducts produce general rules, while large reducts produce specific rules. Various factors can cause the outcome to not be completely discernible by the inputs. These include cases where there is noise in the data, the input attributes do not completely describe the outcome, complex relationships exist between the input and decision attributes, or the input attributes are divided non-optimally into discrete categories.

Figure 1 shows an example of a two input case where the outcome is either a 'yes' or a 'no'. Each input is divided into six discrete categories, and it is assumed that no noise is present on either the inputs or the outcome. The region enclosed by the irregularly shaped loop marks the true boundary between 'yes' and 'no' outcomes for different combinations of the two inputs. The white boxes inside the loop are called the positive region, or lower approximation of the outcome. These are boxes that always have 'yes' outcomes in the table of examples. The negative region, where all examples have an outcome of 'no', is the combination of all white boxes outside of the boundary loop. In between these two areas is a boundary region, marked by gray boxes in Figure 1. In the boundary region, some examples will have a 'yes', while others will have a 'no' outcome. Because the examples that fall in the boundary region are inconsistent, two types of rules are often generated by rough set algorithms, called certain and possible rules. Certain rules are developed from a set of completely consistent examples (e.g., the positive and negative regions in Figure 1), while possible rules are generated from a set of inconsistent examples (e.g., the boundary region in Figure 1). As will be shown, however, because rules are generated from a limited sample of examples, there are no rules that are completely certain.

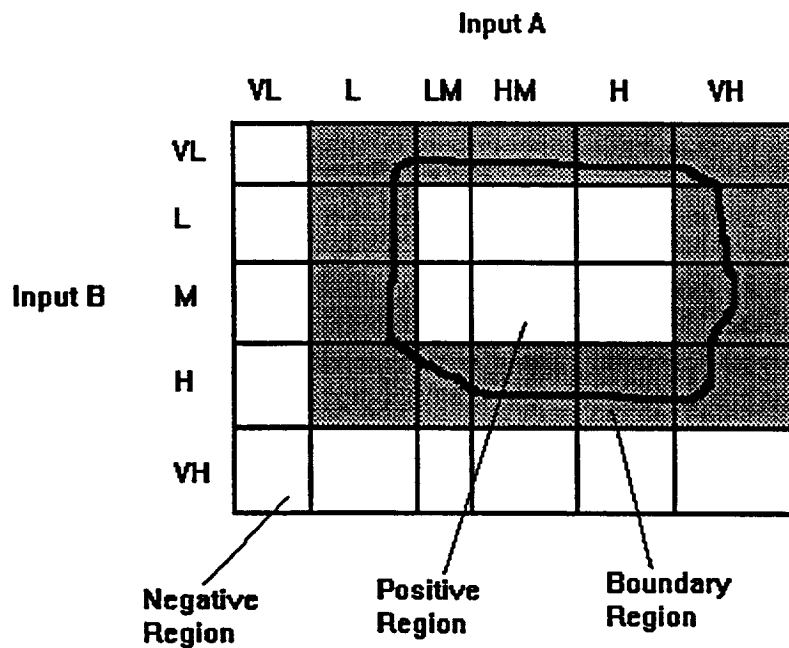


Figure 1.- Rough set example with consistent data.

The boundary region in Figure 1 can be made smaller by changing the definition of the attribute categories for the inputs. For example, if the definition of VL is increased for input A, the boundary region will decrease in size. The price to pay for such a move is that rules generated for input A being L will have fewer examples supporting them, and will thus have less certain validity.

Figure 2 shows the same case as in Figure 1, except that now noise has been added to the data. The noise adds inconsistencies to what should be the positive and negative regions of the model, and sometimes creates a false consistency in the boundary region. If the rough set model is generated from a large set of examples, this will not cause a major problem because the patterns will be clearly discernible through the noise. A difficulty arises, however, when the number of examples the model is based on is small in any region of the input space. Because the number of examples for a given combination of inputs is small, noise can significantly alter the apparent outcome for that combination.

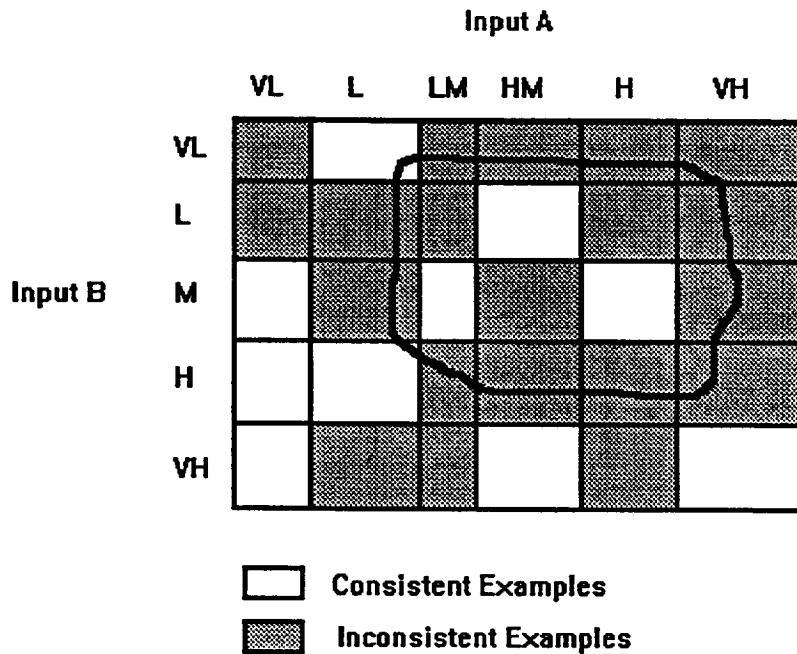


Figure 2.- Rough set example with noisy data.

There are many reasons a data set might have relatively few examples in certain regions. Certain values of an input may occur infrequently, or it may be physically difficult to collect data for these input ranges. Sometimes, it is desirable to predict an event that occurs only rarely, such as a moderate to severe earthquake in Arkansas. Often, a model will have a large number of inputs, making the number of examples required to fill all possible combinations of inputs quite large. A more common difficulty is when predictions

need to be made on dynamic systems. For example, cardiac patients benefit from a large number of drugs that become available each year. Conditions that would have been fatal a few years ago are now readily treatable, and any model that predicts cardiac mortalities needs to be continually revised using only data from recent patients.

When rules are generated based on a few examples, the question arises as to how much confidence one can have in these rules. This question becomes critical when the penalty for a wrong decision is catastrophic. For example, a model that incorrectly predicts the presence of cancer in a patient may cause needless worry and extra expense for additional tests, but if that model incorrectly predicts the patient has no cancer, the results could be deadly. For decision attributes with only two possible values (e.g., yes or no), the confidence one has in a given rule can be determined by examining the binomial probability distribution. For decision attributes with more than two values, the analysis below also holds true if one only wants to know the probability of a given rule being correct.

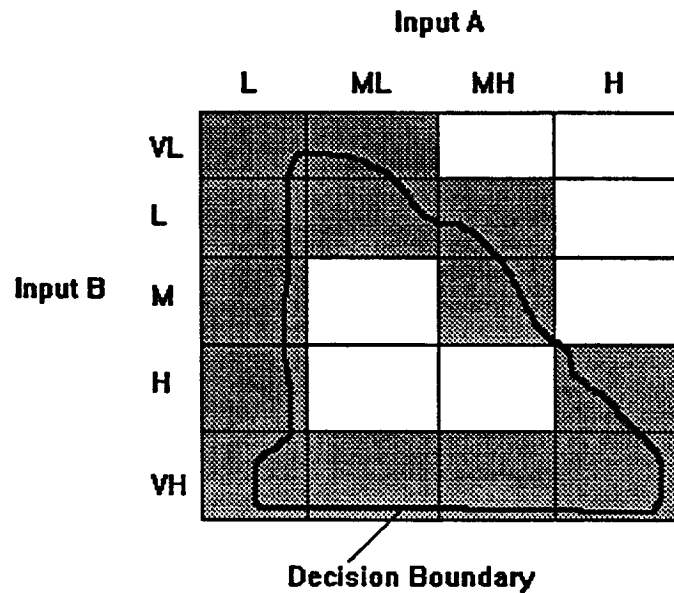


Figure 3.- Example of how contradictory rules are generated.

Another feature of rough sets is that contradictory rules are often created in an attempt to have the strongest, most general rules possible. To see how this occurs, look at Figure 3. Recall that the fewer variables in a rule, the more general it is. In this case, there are two inputs and an output that is either a 'yes' or a 'no'. The decision boundary between these two states is shown by the irregular-shaped loop. Suppose we are willing to accept any rule that is correct over 70% of the time in an attempt to have a very general system. In Figure 3, when Input A is L, the output will be 'no' more than 70% of the time. When Input B is H, the output will be 'yes' more than 70% of the time. With these two rules, a conflict will occur whenever Input A is L and Input B is H at the same time. A method of selecting the strongest rule is necessary to resolve this conflict. The next

section presents a method of determining the confidence level one can have in a given rule, which can then be used to resolve conflicts between rules.

PROBABILITY CALCULATION

Given n samples from a binary distribution, r of which are true, what is the probability p of a true response? If p is known, then the probability $P(n, r, p)$ of getting r true responses is:

$$P(n, r, p) = \binom{n}{r} p^r (1 - p)^{n-r} \quad (1)$$

To find the mode, the peak of the distribution, simply maximize (1) with respect to p as follows:

$$\frac{d}{dp} P(n, r, p) = r p^{r-1} (1 - p)^{n-r} - p^r (n - r) p^{n-r-1} = p^{n-r-1} (r(1 - p) - (n - r)p) = 0 \quad (2)$$

Solving (2) one gets

$$r - rp = np - rp \quad (3)$$

or $p = r/n$. To obtain a probability distribution from (1) one need only normalize as follows

$$D(p; n, r) = \frac{\binom{n}{r} p^r (1 - p)^{n-r}}{\int_0^1 \binom{n}{r} p^r (1 - p)^{n-r} dp} \quad (4)$$

The integral in the denominator of (4) can easily be evaluated using integration by parts.

Let $I(n, r) = \int_0^1 p^r (1 - p)^{n-r} dp$. Then

$$I(n, n) = \int_0^1 p^n dp = \frac{1}{n+1} \quad (5)$$

and

$$\begin{aligned} I(n, r) &= \int_0^1 p^r (1 - p)^{n-r} dp = \frac{1}{r+1} p^{r+1} (1 - p)^{n-r} \Big|_0^1 + \frac{n-r}{r+1} \int_0^1 p^{r+1} (1 - p)^{n-(r+1)} dp \\ &= \frac{n-r}{r+1} I(n, r+1) \end{aligned} \quad (6)$$

Solving the recurrence (6) with end condition (5), one gets

$$I(n,r) = \frac{1}{n+1} \prod_{j=r}^{n-1} \frac{n-j}{j+1} = \frac{1}{(n+1)\binom{n}{r}} \quad (7)$$

The mean of the distribution (4) is realized as

$$\bar{p} = \frac{I(n+1,r+1)}{I(n,r)} = \frac{(n+1)(r+1)!(n-r)!n!}{(n+2)r!(n-r)!(n+1)!} = \frac{r+1}{n+2} \quad (8)$$

Note that the expected value of p (the mean value of p) is different from the mode (the peak value of the distribution) for all cases except $r/n = 0.5$.

Confidence limits for the underlying value of p can be obtained by numerically integrating (4) with respect to p for any given values of n and r . Starting with $p = 0$ and increasing it incrementally, the area of the distribution from 0 to any given value of p can be calculated. Since the total area of the distribution is equal to one, confidence limits can easily be placed on upper and lower approximations of p . For example, if the area under (4) for a given value of r and n is 0.05 from $p = 0$ to 0.60, then there is a 5 percent chance that the true underlying probability for this case is less than 0.60.

NUMERICAL MODEL VERIFICATION

The above analysis answers the question of, given r/n 'yes' responses, what is the likely underlying probability p of the system? In order to numerically verify the results, this question must be rephrased to: given a known underlying probability p , what is the likelihood of getting r 'yes' responses out of n trials? An algorithm was written to answer the latter question for $n = 10$ and $r = 5$ through 9. Probabilities were incremented from 0 to 1 in 0.001 steps, with 40,000 trials were run for each probability. In each trial, ten numbers were randomly generated and the results were checked for r 'yes' responses. The total number of $r/10$ 'yeses' was recorded for each p , and the area under the distribution was calculated from 0 to p for each probability. The results were then compared with the above analysis for calculating the mean, the mode and confidence limits on the underlying probability for any given observation of $r/10$ 'yes' observations.

Results for the theory based calculations can be compared to the ones from the numerical experiment. The largest differences between the numerical and analytical results occurred for $r = 9$. In this case, the maximum difference between the cumulative distributions is 0.0019, or less than 0.2% of the maximum. The 10% confidence level is 0.69 and the 90% confidence level is 0.95 in both cases. When the cumulative distribution for the experimental case is plotted versus the theory-based calculations, the result is a line with a correlation coefficient (Pearson's R) of 1.000. The mode for both cases is 0.9, which equals the expected value of n/r .

DISCUSSION

The most obvious conclusion from the above analysis is that there are no certain rules in rough sets. Rules are generated based from a sample of similar cases, and it is hoped that those cases are representative of all cases the rough set model will ever see. The typical probability normally used to evaluate rules, $p = r/n$, represents the peak of the distribution that represents all possible values of p for a given combination of r and n . The distribution, however, is skewed to the left for all values of $r/n > 0.5$. The minimum error over large number of rules will therefore occur for the probability that divides the distribution into two equal areas, $p = (r+1)/(n+2)$, rather than r/n . Even with a completely consistent set of examples, the minimum error over a large number of rules will always occur by choosing $p < 1$, irrespective of the number of consistent examples supporting the rule. For example, if a person flips a coin three times in a row, on average they will come up with three heads once in eight times. Having seen a person flip this coin three times, all of which came up heads, one cannot assume that the coin will always come up heads. Now suppose in a given rough set model there are numerous combinations of inputs that all have three examples in them, all with 'yes' responses. The average probability of getting a 'yes' response, over all these rules is 0.8, not 1.

Sometimes it is desirable to have a rule used only when it has a high probability of being correct. Confidence limits can be used to determine what rules to use. Note, however, that high confidence limits require a large number of examples. For example, if one wishes to have a 99% confidence that $p > 0.9$ for a rule to fire, it requires a minimum r/n of 43/43, 62/63, or 78/80. Other times it is a necessary to make a "best guess" as to what the correct answer is, in that case the decision indicated by r/n is always the most appropriate choice.

Often with rough set models it is necessary to decide between conflicting rules. The mean probability, $\bar{p} = (r+1)/(n+2)$ can be used to determine which rule is stronger. Often the cost for a wrong decision is greater than the reward for a correct decision, such as in the case of screening for cancer. Mean probabilities can be used with a cost functional to determine the optimal decision. Let tp represent a correct 'yes' predictions for a rough set model, tn represent a correct 'no' prediction, fp an incorrect 'yes' prediction and fn an incorrect 'no' prediction. A cost function, C , can be defined as follows:

$$C = a_1 tp + a_2 tn - a_3 fp - a_4 fn \quad (9)$$

where $a_1 - a_4$ are the relative costs of each decision. Let p_y be the mean probability of a true positive (tp), $(1 - p_y)$ the mean probability of a false positive (fp), p_n the mean probability of a true negative (tn) and $(1 - p_n)$ the probability of a false negative. For a positive rule to fire, $a_1 p_y - a_3 (1 - p_y)$ must be greater than 0; conversely, $a_2 p_n - a_4 (1 - p_n)$ must be greater than 0 for a negative rule to fire (otherwise a no decision is better). To

choose between conflicting positive and negative rules, the benefits of each rule must be compared. For a 'yes' rule to have more advantage than a 'no' rule,

$$a_1 p_y - a_3 (1 - p_y) > a_2 p_n - a_4 (1 - p_n) \quad (10)$$

Solving for p_y , we get

$$p_y > (a_2 + a_4) / (a_1 + a_3) * p_n + (a_3 - a_4) / (a_1 + a_3) \quad (11)$$

for a 'yes' rule to control the decision, otherwise the 'no' rule prevails.

The analysis in this paper was carried out for a decision attribute with only two possible values. It is easy to see that this can be expanded to any number of categories. If one is only interested in whether a decision is correct or not, then the analysis holds as is. Otherwise, the probability description given in (1) can be modified to include more than two states, and the calculations redone.

APPLICATIONS

The rough set methodology was applied to two applications, detection of the space shuttle electronic "signatures" and prediction of decompression sickness. The space shuttle has numerous electrical systems that periodically turn on and off. It is often desirable to know when individual appliances are operating, but the number of these devices makes using telemetry to send this information back to earth problematic. The electrical power usage in the shuttle, however, is regularly monitored by mission control. Fluctuations in the power indicate that different apparatus is turning on and off. In the past, human operators examined these fluctuations to determine which piece of equipment was turning on and off. Automation of the electrical power "signatures" recognition would eliminate the need for human monitors.

Rough sets was used to classify ten different electrical power signatures, including a "none of the above" category. On a model development set, the rough sets technique was able to correctly classify 84.6% of over 5,000 cases, with 14.6% no decisions and 0.9% missclassified. When electrical signatures in the "none of the above" category were removed, the rough set model correctly classified 98.7% of the signatures, with a no decision in 1% and 0.3% missclassified. These preliminary results indicate that additional work on rough set model development for electrical "signature" recognition is justified.

Decompression sickness (DCS), commonly known as the "bends", occurs when people experience rapid changes in external pressure. This most commonly occurs when divers resurface too rapidly from deep dives, but may also happen when pilots fly at high altitudes. Astronauts performing an extravehicular activity (EVA), or space walk, may also be at risk of developing DCS due to the low pressure in space suits. A preliminary rough set model of decompression sickness was developed from retrospective data on

1018 exposures. Fourteen physiologic and case history inputs were available for each subject, and the outcome was the development or absence of DCS. The model was tested on 706 subjects that were not among the 1018 cases used for model development. The rough set model was able to correctly classify subjects as experiencing DCS or not 83% of the time. This compares well with the 79% correct classification seen with stepwise logistic regression, and warrants further investigation of the rough set model.

REFERENCES

- Chan, C.C., "Incremental Learning of Production Rules from Examples Under Uncertainty: A Rough Set Approach", *Int. J. Software Eng. and Knowledge Eng.*, vol. 1, No. 4, 1991, 439-461.
- Grzymala-Busse, J.W., "Knowledge Acquisition Under Uncertainty: A Rough Set Approach", *J. Intel. Robotic Syst.*, vol. 1, 1988, 3-16.
- Pawlak, Z., "Rough Sets", *Int. J. Computer Inf. Sci.*, vol. 11, 1982, 145-172.
- Pawlak, Z., Wong S.K.M., and Ziarko, W., "Rough Sets: Probabilistic versus Deterministic Approach", *Int. J. Man-Machine Studies*, vol. 29, 1988, 81-95.
- Slowinski K., and Slowinski, R., "Sensitivity Analysis of Rough Classification", *Int. J. Man-Machine Studies*, vol. 32, 1990, 693-705.
- Slowinski, R., "Strict and Weak Indiscernibility of Objects Described by Quantitative Attributes with Overlapping Norms", *Foundations of Computing and Decision Sciences*, vol. 18, 1993, 361-369.
- Szladow, A., "Rough Sets: Working with Imperfect Data", *AI Expert*, vol. 8, 1993, 36-41.