HIGH RANGE RESOLUTION RADAR TARGET CLASSIFICATION:

A ROUGH SET APPROACH

A Dissertation Presented to

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology

Ohio University

In Partial Fulfillment
of the Requirement for the Degree

Doctor of Philosophy

by

Dale E. Nelson

June, 2001

THIS DISSERTATION ENTITLED

“HIGH RANGE RESOLUTION RADAR TARGET CLASSIFICATION:

A ROUGH SET APPROACH”

by Dale E. Nelson

has been approved

for the School of Engineering and Computer Science

and the Russ College of Engineering and Technology

Dr. Janusz A. Starzyk, Professor

Dr. Jerrel Mitchell, Interim Dean
Fritz J. and Dolores H. Russ
College of Engineering and Technology

Acknowledgements

This dissertation is the culmination of 20 years of effort. It could not have been accomplished without the help and support of many people. I first want to thank my wife, Chris, for sticking through the many years of disappointments and hard work. I thank her giving up the many hours we could have been together while I performed the work required to complete this research. I especially appreciated the many hours she spent riding from Dayton to Athens every two weeks and the hours spent waiting as I consulted with my advisor. I do not know if I could have “stuck with it” without her quiet (and sometimes not so quiet) encouragement and support.

I thank the United States Air Force for the financial sponsorship of this work. I appreciate all the many hours that were permitted by various supervisors (especially Mr. Jerry L. Covert) and their encouragement along the way.

Finally, to one of the most important people, I thank my advisor, Dr. Janusz A. Starzyk, who encouraged me to initiate the effort at Ohio University and then saw me through it. During this time not only was he an advocate and advisor, but most of all, a friend. There are no words that can express my gratitude to this man. He is, without a doubt, the finest professor I have ever had!

Table of Contents

List of Tables

List of Figures

1 Introduction

The United States Army, Navy, Air Force, and the intelligence community have all been actively researching the subject of Automatic Target Recognition (ATR) algorithms for many years. Part of the reason for this research has been the explosion in the amount of information that is becoming available. The United States is bringing on line more satellites, sensors, and sensor systems which are responsible for this deluge of information. At the same time there has been a push to downsize government in general and the military specifically. This means that there are far fewer people to interpret the data. To make matters even worse the time available to make critical decisions based on the data is becoming shorter. As the data becomes more complex with new sensors pushing the envelope of knowledge and with fewer people available to develop the new generations of ATR systems, scientists and engineers have looked to computers to assist with this development. Innovative sensors capable of sensing new and broader aspects of the electromagnetic spectrum are befuddling past methods of interpreting this new “imagery.” In the past human intelligence, experience, and intuition were excellent guides. However as these new sensors move more toward sensing DC to daylight, intuition fails. Machine learning provides a new hope that computers can develop new ways of interpreting data and transforming it into information.

There is a hierarchy associated with converting data into knowledge or understanding [34].

Data - Data is at the bottom of the hierarchy and normally consists of raw numbers. This could be the signal strength values in range bins or latitude and longitude numbers of a target’s location.

Information - In order to turn data into information the data must be associated, organized, and perhaps fused with other data. Once this is accomplished trends become apparent. These trends tell what is happening at a basic level. A series of latitude and longitudes of a given target indicate direction and speed.

Knowledge - Information becomes knowledge when it is combined with context, education, and experience. Knowledge indicates why something happened giving a better view of a bigger picture. Knowing the direction and speed of a target along with the target's ID and similar movements from other targets may reveal the mission of the target.

Understanding - Understanding is at the top of the hierarchy. When knowledge and intuition are combined with perspective and judgement obtained over time understanding is developed. Understanding permits anticipation of future information, how it fits in the overall picture, and what actions should be taken based upon it.

It is the goal of machine intelligence and data mining to turn data into knowledge and perhaps understanding. ATR's goal is to take the myriad of data being spewed forth by a multiplicity of sensors and convert this into knowledge to be used by commanders and troops to effectively accomplish their mission.

1.1 Background

Although not a new sensor, High Range Resolution (HRR) radar is becoming increasingly more important as an ATR sensor. New radars are being deployed with this mode of operation. This sensor collects data which is a range profile of an aircraft. The data obtained is the result of the electromagnetic scattering characteristics of the target as a function of the range along the line of sight of the radar. Strong returns will come from items such as the canopy-fuselage interface, the nose cone, the wing root, the tail root, and engine cavities to name just a few. The goal of the ATR system is to identify the target (aircraft type in the case of this research) from this data. The concept of **HRR radar** is illustrated in Figure 1-1.

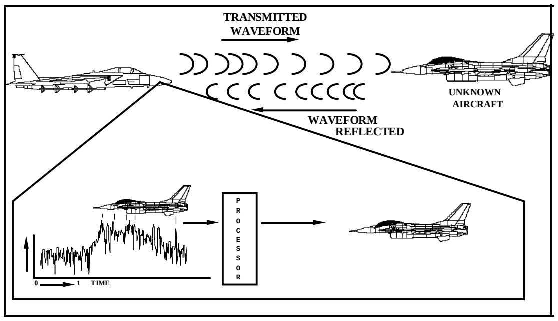


Figure 1-1 HRR RadarTarget Identification

HRR radar target identification is not an easy task. Figure 1-1 shows that a three dimensional object is being represented as one dimensional signal. Three dimensions are collapsed into one. The radar signals are typically modeled as complex exponentials [22]. When these are combined during the dimensional reduction, they add constructively or destructively depending on their relative phases. Therefore just a slight change in the relative phases in the returns can have significant effect on the signature. This results in a high degree of variability in the target signature. This variability can easily be seen in Figure 1-2 which contains HRR returns over a one second period. Two aircraft are shown with 5 observations, 200ms apart. During this time period the azimuth and elevation changed by less than one degree. This variability in signature is characteristic of all HRR signatures. Due to this variability most ATR approaches, until now, have been statistically based.

For small numbers of targets the targets have been shown to exhibit good separability using standard statistical pattern recognition techniques [23]. The problem has typically been meeting performance requirements. Typically these include:

a) high probability of declaration (P_{dec}).

b) high probability of correct classification (P_{cc}).

c) low probability of misidentifying an unknown target.

Known targets are those that are contained in the training data set. Unknown targets are anything that is not in the training data set. For an ATR system to be useful it must declare, when presented with a known target, a high percent of the time. When the system does declare, it must be correct almost all the time. If the target is unknown, the system should either declare that or just not make any declaration at all. These are conflicting goals. If the ATR system does not have to declare often, then it is easy to just declare when the system is absolutely certain. The P_{cc} will be very high but P_{dec} will be very low. Conversely if the system is forced to declare almost all the time, it likely will be wrong a significant part of that time. The unknown target classes are problematic. Obviously they cannot be included in the training data set or else they would be known targets. To put this in perspective, a system requirement might be $P_{cc}> 90\%$ and $P_{dec}> 85\%$. These figures would be contained in the specifications for a given system based on its mission.

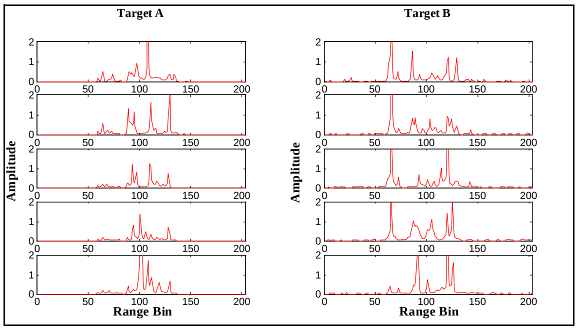


Figure 1-2 One Second Sequence of HRR for Two Aircraft

A HRR is considered a nonliteral sensor. That is it does not match signals humans are familiar with. A person seeing a picture can easily do recognition on it because it is the same as normal vision. A HRR signature would be better interpreted by a person if it was listened to (sound is a one dimensional signal). However, the HRR signature is presented as a visual signal and humans are not very good at interpreting it. Therefore, intuition does not provide insight in how to devise an ATR system for HRR. The focus of this research is to apply advanced machine learning to creating an ATR capable of meeting operation requirements for the Air Force.

1.2 Related Research

The literature reviewed on classifiers for identifying HRR signatures for this research effort covered the most popular classification technique, the constrained quadratic classifier, and one of the most recent developments, the Statistical Feature Based Classifier with Feature Level Fusion (StaF).

Current approaches for HRR use the entire range profile as a feature vector. The training of this classifier is a statistical parameter estimation problem. For better accuracy, due to the extreme variability of a target signature with azimuth and elevation, classifiers are typically developed for a small viewing window of approximately 5^o in azimuth and 5^o in elevation. This is illustrated in Figure 1-3.

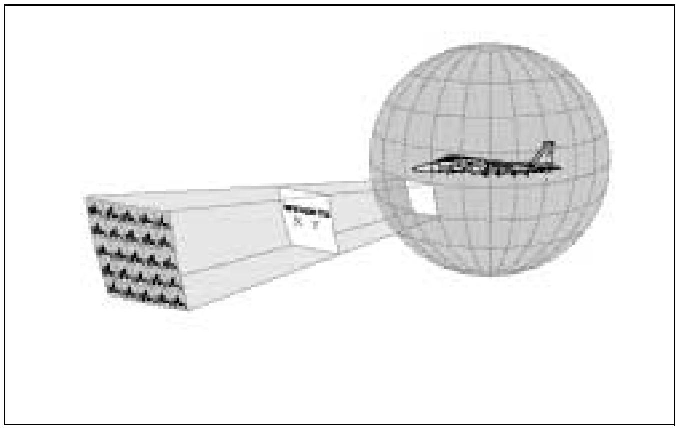


Figure 1-3 Windowed Partitioning of Data

The parameters to be estimated will depend on the algorithm to be used.

1.2.1 Quadratic Classifier

One of the most frequently chosen techniques for classification of HRR signatures is the **constrained quadratic classifier** [24]. This classifier is based on computing the mean and variance for each range bin in the signal. The discrimination function for this classifier is:

$$h_k(x)= \sum_{i=1}^{N} \frac{(x_i-\mu_{ik})^2}{\sigma_{ik}^2} + 2\sum_{i=1}^{N} \ln\sigma_{ik} \tag{1-1}$$

where x_i, μ_{ik}, and σ_{ik} are the observation, mean, and variance of the i^{th} range bin for target class k. The classification of the observation is determined from the maximum likelihood criteria,

$$h_j(x) = \min_k h_k(x) \ \rightarrow x \in \omega_j \tag{1-2}$$

where ω_j is the maximum likelihood target class. Another variant of this technique is to use mean square error which does not use the variance term. This class of target recognizer works best when the number of target classes is small (5 to 10 targets). This approach does very poorly at rejecting or not declaring on unknown target classes. Further, it is not robust due to the fact that it tries to match range bins in the signal which contain little or no information about the target. Typically these are the range bins at the beginning of the signal and near the end of the signal.

1.2.2 Statistical Feature Based Classifier with Feature Level Fusion (StaF)

Mitchell, in his dissertation [22] introduced the **Statistical Feature Based Classifier with Feature Level Fusion (StaF)** to solve some of the problems associated with the constrained quadratic classifier. In Mitchell's classifier a sixth order Daubechies mother wavelet [Table 5-1, Db6] is used to provide noise cleaning of the signal. The features are extracted using of an auto regressive filter and selecting features using peak amplitude thresholding, as Mitchell says "on the fly", making no assumptions on the location or number of features. The features are the location and magnitude of peaks in the signal. Mitchell then uses a combination of Bayesian and Dempster-Schaeffer evidence theory to accumulate evidence resulting in classification.

There are several problems with Mitchell's approach. First is that several probability density functions (PDFs) are required in order for this classifier to work. Specifically, the peak location probability function (PLPF) and the peak amplitude probability density function (PAPDF). These are then used to calculate the class likelihoods and the class *a posteriori* probabilities for individual extracted features. The PLPF is estimated from the training data ensemble using a Parzen estimator. The PAPDF assumes that the distribution within a given range bin is Gaussian. Mitchell says that the distribution is known to be Rician, however, the Gaussian assumption is reasonable if a power transform is performed. Then in order to increase performance Mitchell uses a five look sequence to achieve his results.

The research presented here is quite different from Mitchell's work. First, this research uses a Haar wavelet to enrich the feature space, not for noise cleaning. Second, no assumption is made concerning the PDF of the data. The probabilities are computed directly from the training data. Third, the data is not partitioned into an ensemble of 5^{o} windows for classification purposes, there is only one large window so no pose estimation is required. Fourth, only one snapshot of the target is required, not a five image sequence. Last, Mitchell required the use of 50% of his available data for training. This research uses 25% for training and 75% for testing.

1.3 Research Objectives

The author has been working in the area of machine intelligence for over twenty years. He has been leading a research group in the area of automatic target recognition for over twelve years. He is very familiar with current research techniques. During this tenure it has become apparent that there are many areas which could be improved upon, especially in the area of statistical pattern recognition as applied to ATR of HRR signals. The most glaring problem is the assumption or estimation of PDFs. Applying emerging machine intelligence and data mining techniques to overcome these estimations and assumptions in current statistical classifiers is highly desirable. A relatively new approach to data mining was introduced by Pawlak [29]. Although this method has been around since the early 1980s it has not received much notice in the United States. This new theory has the potential to produce a more robust classifier. rough set theory assumes that the training set is all that is known and all that needs to be known to do the classification problem. Techniques to find the minimal set of attributes (range bins) to do the classification are available in the theory. Further, since the theory will find all the classifiers, the system should be robust.

The primary objective of this research is to develop a workable, robust classification methodology using machine learning and data mining techniques. Specifically the approach should

- generate features for the purpose of classification.
- determine which features are important.
- generate a multiplicity of classifiers.
- determine a method of fusing classifiers for robustness.
- be computationally appropriate for deployment.

Once the data is labeled rough set theory guarantees that all possible classifiers using that training data set will be generated! There is no equivalent statement that can be made using statistical pattern recognition. However, generating all the classifiers has been shown to be a NP-hard problem [40]. Therefore, this research had a secondary objective to find ways to overcome this problem and make rough set theory (RST) usable in real world size problems.

One of the strongest requests for this research comes from [32] where Pawlak states in the foreword:

> It is especially important to develop widely accessible, efficient software for rough set based data analysis, particularly for large collections of data. Despite of many [sic] valuable methods, based on rough set theory, for efficient generation of optimal decision rules from data, developed in recent years, more research is needed here, particularly, when quantitative attributes are involved. In this context also new

> discretization methods for quantitative attribute values are badly needed.

1.4 Dissertation Outline

This dissertation provides background material on rough set theory in Chapter 2 and feature selection techniques and fusion in Chapter 3. Chapter 4 introduces the data preprocessing required to develop a classifier and presents original research on how to preform the labeling task required by RST. In addition, original research is introduced adding fuzziness to the test data to improve classifier performance. Chapter 5 introduces wavelets in the context of enhancing the feature space of a classifier. Original research determining which wavelet to use and a new concept of an iterative wavelet is presented here.

Chapter 6 discusses ways to reduce the dimensionality of the data to allow RST to be applied. This requires introducing some new definitions to extend RST. In addition, original research on partitioning the data to produce more robust classifiers is introduced along with a new fusion technique for combining the resulting classifiers. One of the more important original research contributions is discussed, a method of determining near minimal reducts, which allows RST to be applied to real world size problems.

The dissertation provides the results of computer simulations in Chapter 7 confirming that the postulated theories work together to produce a robust rough set classifier. This chapter also shows that the near minimal reduct is close to minimum size (number of attributes).

The dissertation concludes with suggestions for further studies and research to determine the limits of improvements possible with the new methods and theory introduced.

An appendix is included which completely defines all the reducts and their associated core for each of the partitions. This appendix provides data which can give the reader insight into the results summarized in the tables, graphs, and narratives of the dissertation.

2 Rough Set Theory

A great deal of time and effort has been used by companies and governments since the advent of computers to acquire and store data in large repositories. It is easy enough to see that this has been a strong trend by just looking at the evolution in storage media. When the first IBM PC was introduced in the early 1980s it came with a cassette interface and it was possible to get 5.25 inch floppy disks which stored 160K bytes of information. When the first hard disk was introduced for this same computer, it would hold five megabytes of information. People would talk about what could one possibly put on the disk that would require that much space. Now in 2001 disk drives are readily available for the PC which will hold 80 gigabytes. As more data has become available, the storage media has increased in size and reduced in cost to accommodate this information. The military is increasingly worried about a problem known as the pixel to pupil ratio. This problem relates to the fact that there is now a capability of receiving and storing data at a rate that is greater than the ability of analysts to interpret. It is the interpretation that turns the data into knowledge.

rough set theory was developed by Zdzislaw Pawlak in the 1980s [29] as a formal method to turn data into knowledge. This is also termed **data mining**. The basic premise is that the data in the training set is all that one knows about the problem at hand. All that can be observed is contained in the data set. Turning data into knowledge is not an easy task. The data will likely be redundant and contain far too much detail to be useful for decision purposes. For example, some attribute values may come from the real numbers. If these attributes are used as is, it will be difficult to find patterns and make groups of similar classes. The data needs to be quantified or labeled. Often rough sets are compared to fuzzy sets. A simple analogy is the easiest way to illustrate the difference. If one thinks about set elements as picture elements (pixels), fuzzy sets are concerned with how gray a pixel is while rough sets are focused on the size of the pixel [29]. The idea here is that if one can only describe the aircraft shape in Figure 2-1 in terms of the pixels on the left, the shape would not be as well defined as it would using the pixels on the right. This concept is used in determining the labeling discussed in Section 4.4.

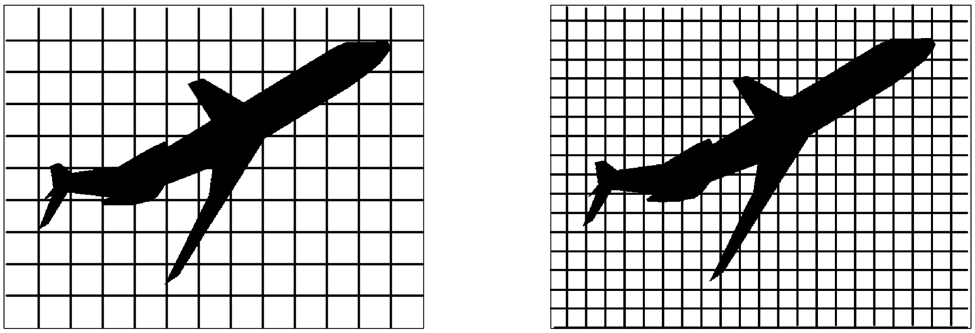


Figure 2-1 Pixel Size Effect

rough set theory has features applicable to the classification problem. Some of these features are:

1. it is an algebraic method.
2. it is applicable to problems with both numeric and descriptive attributes.
3. it is capable of finding all minimal knowledge representations.
4. it is highly automated based on strict rules.
5. it is discrete (unlike fuzzy computing).
6. it is applicable to statistical as well as rule based learning.
7. it is robust.
8. it is normally limited to small size problems.

Some of the features listed are problematic with real world sized problems. This research presents a methodology to exploit the strengths and resolve the problems.

2.1 Information Systems

Frequently data is stored as a relational database. This takes the form of a table where the rows are records and each column contains some observed information about the object associated with the record. In rough set theory this is called an **information system** (U,A) where $U = \{x_1, \ldots, x_n\}$ is a non-empty finite set called the **universe**, and $A = \{a_1, \ldots, a_n\}$ is a non-empty set. The elements of A, called **attributes** (in our case range bins), are functions

$$a_1 : U \to V_i \tag{2-1}$$

where V_i is called the value set of a_i. In a practical rough set system V_i is a discrete and finite set of values. In the case of binary labeling used in this work $V_i = \{0, 1\}$. Often in an information system there are two different kinds of attributes, one is called a **condition attribute** and the other is called a **decision attribute**. In this case the information system is $(U, A \cup \{d_i\})$ where d_i are the decision attributes. Condition attributes represent what is observable while the decision attributes are things which are to be determined from the observations. For this research the condition attributes are the range bins and the decision attribute is the target class. Frequently it is possible to remove some of the condition attributes and not introduce any ambiguities into the information system. An **ambiguity** is when two rows have identical attribute values but one row is associated with one decision attribute and the other row with a different decision attribute. When condition attributes are removed without creating ambiguities, the information system is said to have been reduced (without loss of information) and the reduced system is said to be a **reduct** (a term that will be formally defined in Section 2.3).

2.2 Definition of Discernibility Matrix

Skowron [40] proposed representing an information system in the form of a **discernibly matrix**. The discernibility matrix is then used to extract knowledge (and reduce it to a minimal form) from the information system by simple algebraic manipulations. The discernibility matrix of A is the n x n matrix with i, j^{th} entry

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\} \tag{2-2}$$

In the most general case each x_i represents a separate class. So an element of c_{ij} of a discernibility matrix contains all the attributes that differentiate between two given objects x_i and x_j. Since the discernibility matrix contains all groups of attributes that differentiate between all objects of the universe (all classes of interest to the information system) it can therefore be used to derive all of the classification rules. It is this feature of the discernibility matrix that is of primary interest in machine learning and classification methods in particular. Let $B \subseteq A$ and let P(A) be the power set of A. The Boolean-valued function χ_B is

$$\begin{aligned} \chi_B &: P(A) \to 0, 1 \\ &: C \mapsto \begin{cases} 1 \text{ when } B \cap C \neq \varnothing \\ 0 \text{ when } B \cap C = \varnothing \end{cases} \end{aligned} \tag{2-3}$$

Let $S_x = \{\chi_B : B \in P(A)\}$. It is desired to write the discernibility function. This is normally written using the Boolean symbols for conjunction and disjunction. However, these symbols do not mean what they mean in the normal Boolean sense. Therefore definitions of these operators must be included so that there is no misunderstanding. Additionally, the associative and distributive property must also be defined. A complete development of this section with mathematical proofs may be found in [45]. Define the binary operator $\wedge$, called **conjunction**, by

$$\begin{aligned} \wedge &: S_x \times S_x \to S_x \\ &: (\chi_B, \chi_C) \mapsto \chi_B \wedge \chi_C \end{aligned} \tag{2-4}$$

where

$$\begin{aligned} \chi_B \wedge \chi_C &: P(A) \to \{0, 1\} \\ &: D \mapsto \chi_B(D)\chi_C(D) \end{aligned} \tag{2-5}$$

The associative property

$$(\chi_B \wedge \chi_C) \wedge \chi_D = \chi_B \wedge (\chi_C \wedge \chi_C) \tag{2-6}$$

allows the dropping of parenthesis without any possibility of confusion; moreover now define $\wedge$ for any finite collection of functions $\{\chi_B\}_{j=1}^{p}$ by recursion

$$\bigwedge_{i = 1, \ldots, j} \chi_{B_i} = \left(\bigwedge_{i = 1, \ldots, p-1} \chi_{B_i} \right) \wedge \chi_{B_p} \tag{2-7}$$

The discernibility function of the information system is:

$$\begin{aligned} f_A &: P(A) \rightarrow \{0, 1\} \\ &: C \mapsto \left(\bigwedge_{\substack{1 \le i, j \le n \\ \chi_{c_{ij}} \ne \bar{0}}} \chi_{c_{ij}} \right)(C) \end{aligned} \qquad \textbf{(2-8)}$$

where “$\bar{0}$“is the constant function

$$\begin{aligned} \bar{0} &: P(A) \rightarrow \{0, 1\} \\ &: C \mapsto 0 \end{aligned} \qquad \textbf{(2-9)}$$

If f_A is an empty conjunction, define f_A to be the constant zero function. This is an uninteresting case therefore assume throughout that f_A is not an empty conjunction.

The condition $\chi_{c_{ij}} \ne \bar{0}$ used in the definition of the discernibility function is equivalent to the condition that $c_{ij} \ne \varnothing$ since

$$\chi_{c_{ij}} \ne \bar{0} \Leftrightarrow \chi_{c_{ij}}(A) = 1 \Leftrightarrow c_{ij} \cap A \ne \varnothing \Leftrightarrow \exists a_k \in c_{ij} \Leftrightarrow c_{ij} \ne \varnothing \qquad \textbf{(2-10)}$$

Using the fact that the discernibility matrix is symmetric and that $c_{ij} = \varnothing$ it follows that the discernibility function simplifies to $f_A = \bigwedge_{\substack{1 \le i < j \le n \\ \chi_{c_{ij}} \ne \varnothing}} \chi_{c_{ij}}$. It is also known [44] that $f_a(A) = 1$.

Let $B \subseteq A$. The **B-indiscernibility** relation is:

$$\mathrm{Ind}(B) = \{(x, y) \in U \times U : (\forall a \in B)(a(x) = a(y))\} \qquad \textbf{(2-11)}$$

The **B-discernibility** relation is the complement of Ind(B) in U x U,

$$\mathrm{Dis}(b) = U \text{ x } U - \mathrm{Ind}(B) \qquad \textbf{(2-12)}$$

The following lemma is an immediate consequence of the definition.

Lemma. Let $B \subseteq A$. Then

$$\mathrm{Dis}(B) = \mathrm{Dis}(\bigcup_{a \in B} \{a\}) = \bigcup_{a \in B} \mathrm{Dis}(\{a\}) \qquad \textbf{(2-13)}$$

Consequently, if $a, b \in B$ and Dis({a})=Dis({b}) then

$$\mathrm{Dis}(B) = \mathrm{Dis}(B - \{a\}) = \mathrm{Dis}(B - \{b\}). \qquad \textbf{(2-14)}$$

Not all knowledge presented in the information system is necessary to describe it. Reduction of knowledge in the information system (which results in generation of reducts) is analogous to mathematical independence of vectors in linear algebra. Reduction of knowledge will be based on the expansion and simplification of the discernibility function. Basic tools for this simplification are the absorption and expansion laws discussed in this section. These tools will be used to produce a specific form of the discernibility function defined here as a simple form.

As before, let $S_\chi = \{\chi_B : B \in P(A)\}$. Define the binary operator $\vee$, called **disjunction**, by

$$\begin{aligned} \vee : S_x &\to S_x \\ (\chi_B ; \chi_C) &\mapsto \chi_B \vee \chi_C \end{aligned} \qquad \textbf{(2-15)}$$

where

$$\begin{aligned} \chi_B \vee \chi_C &: P(A) \to \{0, 1\} \\ &: D \mapsto \begin{cases} 1 \text{ if } \chi_B(D) = 1 \text{ or } \chi_C(D) = 1 \\ 0 \text{ if } \chi_B(D) = 0 \text{ and } \chi_C(D) = 0 \end{cases} \end{aligned} \qquad \textbf{(2-16)}$$

It is easy to prove that the operator $\vee$ satisfies associativity, commutativity, and distributes with respect to conjunction,

$$\chi_B \wedge (\chi_{C_1} \vee \ldots \vee \chi_{C_k}) = (\chi_B \wedge \chi_{C_1}) \vee \ldots \vee (\chi_B \wedge \chi_{C_k}) \qquad \textbf{(2-17)}$$

Likewise it distributes with respect to disjunction,

$$\chi_B \vee (\chi_{C_1} \wedge \ldots \wedge \chi_{C_k}) = (\chi_B \vee \chi_{C_1}) \wedge \ldots \wedge (\chi_B \vee \chi_{C_k}) \qquad \textbf{(2-18)}$$

These last two properties are called the **distribution laws**.

Using these two special operators it is now possible to define methods to determine some important properties of rough sets.

2.3 Definition of a Reduct

In any information system it may be possible to discern all the objects in the system without using all the condition attributes. In other words, some of the columns may be eliminated and it is still possible to "classify" all the classes in the system. Generally what is desired is to find the minimal number of condition attributes that maintain all the information in the information system. Essential for the information system are the reducts that describe knowledge represented in the system. Since a single reduct can differentiate between all distinguishable elements of the universe it effectively represents a set of independent rules for classification of objects in the knowledge system. A set $B \subseteq A$ is a **discern** in A if Ind(B) = Ind(A). A discern is called a **reduct** if $(\forall a \in B)\mathrm{Ind}(B - \{a\}) \supset \mathrm{Ind}(B)$, where " $\supset$ "denotes a proper subset relation. This means that a reduct is a minimal set of rules for classification and if any rule is removed from a reduct some ambiguities in classification will result. The set of all reducts of A is denoted Red(A). Thus, if one has Red(A) one can derive all possible minimum classifiers and their classification rules. The reduct generation procedure developed in [40] is based on the expansion of the discernibility function into a disjunction of its prime implicants by applying the absorption and multiplication laws. This procedure is not sufficiently efficient to allow its use in real world size problems and will be subject to modification in this dissertation to make it more efficient.

2.4 Definition of a Core

The **core** of the information system is defined as a set $P \subseteq A$ such that

$$P = \bigcap_{B \in \mathrm{Red}(A)} B \qquad \textbf{(2-19)}$$

In simple terms the core is the most essential part of a reduct. A core consists of attributes that are common to <u>all</u> reducts. If the core is empty, this means that no single attribute is critical for classification. Since the core must be included in all classifiers it is quite often selected first when generating the set of minimal reducts.

2.5 Reduct Determination

There are few methods in the literature for determining reducts. Three of the methods will be described here. It should be noted that finding all the reducts is a NP-hard problem [40]. That is, the solution grows in non-polynomial time as the problem size increases. For real world problems this has made the use of rough set theory impractical. The size of the information systems typically found in the literature have been small (problem size is expressed as number of rows (records) x condition attributes); i.e., 70x4 [30], 100 x 30 [5], 80 x 22 [41], 114 x 27 [51]. Even the largest problems 5416 x 22 and 2130 x 36 [16] are small in comparison to the HRR problem described here 6426 x 1024.

2.5.1 Brute Force

The brute force method uses no sophistication to determine reducts. All possible combinations of condition attributes are tried to see if they are a reduct. This means that there are:

$$\text{Number of Combinations} = \sum_{k=1}^{n} \frac{n!}{k!(n-k)!} \qquad \textbf{(2-20)}$$

where n is the total number of attributes and k is the number of attributes in each potential reduct. It should be noted that the number of combinations to be tried grows extremely (almost exponentially) fast as n increases.

The brute force process of reduct generation is best described through an illustration. Table 2-1 is an example of a typical information system. The column entitled Target ID is a decision attribute and the columns entitled Range Bin 1 - Range Bin 4 are condition attributes. Notice that each row is unique and each condition attribute is a real number. This kind of table is not very useful because it is difficult to generalize and see patterns. Normally we convert real numbers into some meaningful labels. For example, if a person has a temperature of 98.6^{o} a doctor would label that normal. If the temperature is less than 96^{o}, a doctor would label that low. If the temperature was 101^{o}, the doctor would label it a fever. Cars are grouped as sub-compact, compact, mid-size, full size, and sport utility vehicles. This labeling makes it easier to handle data while bringing out patterns and trends. However, it should be noted that once this is done, some discrimination power has been lost. In general this is not critical and is in fact desirable. For the labeling in Table 2-1 the following labeling scheme was chosen: Label = 1 if value is <.25, Label 2 if value is >=.25 and <=.45, and Label = 3 if value is >.45. Table 2-2 shows the results of this labeling. It should be noted that there is no requirement for the labeling ranges to be the same across all columns. Furthermore, there is no requirement for the number of labels to be the same across all columns.

Table 2-1 Raw Dat a

Signal	Target ID	Range Bin 1	Range Bin 2	Range Bin 3	Range Bin 4
1	1	.680	.127	.121	.446
2	1	.948	.248	.022	.440
3	1	.821	.189	.139	.423
4	2	.396	.680	.237	.239
5	2	.441	.851	.184	.239
6	2	.394	.201	.338	.564
7	2	.775	.401	.006	.617
8	2	.241	.359	.412	.773
9	3	.113	.097	.449	.450
10	3	.896	.327	.122	.927

Table 2-2 Labeled Data

Signal	Target ID	Range Bin 1	Range Bin 2	Range Bin 3	Range Bin 4
1	1	3	1	1	2
2	1	3	1	1	2
3	1	3	1	1	2
4	2	2	3	1	1
5	2	2	3	1	1
6	2	2	1	2	2
7	2	3	2	1	3
8	2	1	2	2	3

Signal	Target ID	Range Bin 1	Range Bin 2	Range Bin 3	Range Bin 4
9	3	1	1	2	2
10	3	3	2	1	3

Once the labeling is complete there is the possibility that the resulting table may be ambiguous. The table is said to be **ambiguous** if signals from two different target IDs are the same; i.e., signals are ambiguous if $x_i = x_j$ and $c_i \neq c_j$ where x_i is signal i and c_i is the target ID of signal i. In the case presented here signals 7 and 10 are the same while having different Target IDs. Therefore, both signals are eliminated. The resulting table with no ambiguities is said to be **consistent.** Only a consistent decision table can be a basis for unique classification rules**.** Following this elimination if there are signals which are the same, this information is redundant and only one signal will be retained for computational efficiency. In this example duplicates will not be removed to illustrate equivalence classes.

Table 2-3 Consistent Information System with Duplicates

Signal	Target ID	Range Bin 1	Range Bin 2	Range Bin 3	Range Bin 4
1	1	3	1	1	2
2	1	3	1	1	2
3	1	3	1	1	2
4	2	2	3	1	1
5	2	2	3	1	1
6	2	2	1	2	2
8	2	1	2	2	3
9	3	1	1	2	2

The reducts of the information system can now be computed. Reducts will be determined which preserve all the information in the information system (even reducts which differentiate signals in the same class). This is accomplished by trying all possible combinations of attributes and determining if the signals composed of only the selected attributes maintain a consistent information system. In this example there are 15 possible reducts (Equation

2-20). There are two reducts: Range Bin 1 and Range Bin 2; Range Bin 1 and Range Bin 4. It is obvious that the core (Equation. 2-19) is Range Bin 1. Using these reducts there are five equivalence classes. An **equivalence class** is the set of all signals which are indiscernible from each other i.e., e_1={x_1,x_2,x_3}, e_2={x_4,x_5}, e_3={x_6}, e_4={x_8}, and e_5={x_9} where e represents the equivalence class. Therefore, the target classification consists of all equivalence classes associated with one Target ID (decision attribute); i.e. c_1={e_1}, c_2={e_2, e_3, e_4}, and c_3={e_5}.

2.5.2 Discernibility Matrix

As in the prior example, the data must be labeled. For computational efficiency duplicate signals are removed resulting in the information system represented by Table 2-4. The next step is to construct the discernibility matrix. This matrix is an n x n matrix where n represents the number of signals left after the information is labeled and made consistent, in this case a 5 x 5 matrix. It should be noted that this matrix is symmetric with no entries on the diagonal. Therefore, for conservation of space only the lower triangular portion and its entries will be shown (this will be represented as a 4 x 4 table for clarity) in Table 2-5. As before, the information in the system will be preserved in the reduct determination and classification can be determined from the reducts.

Table 2-4 Consistent Information System without Duplicates

Signal	Target ID	Range Bin 1	Range Bin 2	Range Bin 3	Range Bin 4
1	1	3	1	1	2
4	2	2	3	1	1
6	2	2	1	2	2
8	2	1	2	2	3
9	3	1	1	2	2

In a discernibility matrix any cell with only one entry means that attribute is part of the core. A core is the one attribute (the only attribute) that allows two different signals to be distinguished from each other. Without this attribute some information will be lost which contradicts the idea of a reduct as a minimal set of attributes without information loss. In Table

2-5 there is one entry (signal 9 vs. signal 6) with only one attribute. Therefore, Range Bin 1 is the core. This is the same answer found by the brute force method, Section 2.5.1

Table 2-5 Discernibility Matrix

Signal	1	4	6	8
4	1, 2, 3, 4			
6	1, 3	2, 3, 4		
8	1, 2, 3, 4	1, 2, 3, 4	1, 2, 4	
9	1, 3	1, 2, 3, 4	1	2, 4

To compute the reduct from the discernibility matrix requires the use of the definition:

B is a reduct if

i) $B \cap c_{ij}$ is minimal with respect to inclusion

ii) $B \cap c_{ij} \neq \emptyset \quad \forall_{ij} \neq 0$

Entries in the discernibility matrix will be referred to by the row and then the column labels. For example, entry (6,4) has the set {2, 3, 4}. The procedure then is to select the smallest entry (in this example the entry (9,6) containing only {1}). All other entries (the entry itself is not removed) which contain a 1 are eliminated; i.e., (4,1), (6,1), (8,1), (9,1), (8,4), (9,4), (9,6). This would proceed for entries of size 2 then 3 etc. In the example, at the conclusion of this process there are only two entries left, one containing {1} (9,6) and the other {2, 4} (9,8). The next step is to create the discernibility function by “or-ing” the entries in each cell and then “and-ing” those together (using the definitions of these two operators from Section 2.2.

$$f_A = 1 \wedge (2 \vee 4) \qquad \textbf{(2-21)}$$

Using this definition there are two reducts:

1) Range Bin 1 and Range Bin 2

2) Range Bin 1 and Range Bin 4.

In [40] this is still proven to be a NP-hard problem which means that it cannot be applied to most real world sized problems.

2.5.3 Expansion Algorithm

A new algorithm was devised which would improve upon the time required to compute the reducts using either the brute force method or the discernibility matrix method. The complete development of this algorithm may be found in [44]. Several laws need to be introduced and some definitions need to be made before proceeding.

2.5.3.1 Absorption Law

Let $B \subseteq A$. Suppose $\varnothing \neq C \subseteq D \subseteq A$. If $\chi_C(B) = 1$ then $\chi_D(B) = 1$.

2.5.3.2 Factorization Law

Let $a \in A$ and suppose $\chi_{C_i}(\{a\}) = 1$ for i= 1,...,k.

Then $\chi_{C_1} \wedge \ldots \wedge \chi_{C_k} = \chi_{\{a\}} \vee (\chi_{C_1 - \{a\}} \wedge \ldots \wedge \chi_{C_k - \{a\}})$

2.5.3.3 Expansion Law

Suppose $f_A = \chi_{C_1} \wedge \ldots \wedge \chi_{C_k} \wedge \chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s}$. Let $a \in A$ and suppose $\chi_{C_i}(\{a\}) = 1$ for $i = 1, \ldots, k$, and $\chi_{C_i} = 0$ for $i = k+1, \ldots, s$. Then

$$\begin{aligned} f_A &= (\chi_{C_1} \wedge \ldots \wedge \chi_{C_k}) \wedge (\chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s}) \\ &= ((\chi_{\{a\}} \vee (\chi_{C_{1-\{a\}}} \wedge \ldots \wedge \chi_{C_{k-\{a\}}})) \wedge (\chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s})) \qquad \textbf{(2-22)} \\ &= (\chi_{\{a\}} \wedge (\chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s})) \vee ((\chi_{C_{1-\{a\}}} \wedge \ldots \wedge \chi_{C_{t-\{a\}}}) \wedge (\chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s})) \end{aligned}$$

Letting

$$\begin{aligned} f_1 &= \chi_{\{a\}} \wedge (\chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s}) \\ f_2 &= (\chi_{C_{1-\{a\}}} \wedge \ldots \wedge \chi_{C_{t-\{a\}}}) \vee (\chi_{C_{k+1}} \wedge \ldots \wedge \chi_{C_s}) \end{aligned} \qquad \textbf{(2-23)}$$

the conclusion reads $f_A = f_1 \vee f_2$, where both f_1 and f_2 are conjunctions of the Boolean-valued functions χ. Since each $\chi_B \in S_x$ is a function $\chi_B : P(A) \rightarrow \{0, 1\}$ both f_1 and f_2 are functions $f_i : P(A) \rightarrow \{0, 1\}$. This suggests the following definitions:

A **simple cover** of a discernibility function f_A is a family of Boolean-valued functions $\{f_1,...,f_k\}$ satisfying:

$$f_A = f_1 \vee \dots \vee f_k \qquad \textbf{(2-24)}$$

where each $f_i : P(A) \rightarrow \{0,1\}$ is a conjunction of Boolean-valued functions.

A simple cover $\{f_1,...,f_k\}$ is called a **simple form** of the discernibility function if for each $f_i = \chi_{C_1} \wedge \dots \wedge \chi_{C_{k_i}}$ the indexing sets are pairwise disjoint, $C_i \cap C_j = \varnothing$ for $i \neq j$.

Given any two subsets $B \subseteq A$ and $C \subseteq A$

$$(B \subseteq [a] \text{ and } C \subseteq [a]) \Rightarrow Dis(B) = Dis(C) \qquad \textbf{(2-25)}$$

Two such subsets are called **equivalent**.

Let $B \subseteq A$. If *there exists* an attribute $a \in A$ such that $B \subseteq [a]$ then B is said to be **strongly equivalent**.

Let $\{f_1,...,f_p\}$ be a simple form of f_A. Let $B \subseteq A$ and $C \subseteq A$. If *there exists* an index i and an attribute $a \in A$ such that $B \subseteq [a]_i$ and $C \subseteq [a]_i$ then it is said that *B* and *C* are **locally equivalent**. For any subset $B \subseteq [a]_i$ $(\forall b \in B)$ $\chi_{D_j}(B) = \chi_{D_j}(\{b\})$ for $j = 1,...,k$.

Let $B \subseteq A$. If *there exists* an attribute $a \in A$ such that $B \subseteq [a]_i$, then it is said that B is a **locally strongly equivalent** subset. When it is necessary to emphasize the particular index i, B is said to be a locally strongly equivalent subset of f_i. Two subsets, A and B, can be locally strongly equivalent without being strongly equivalent and vice versa. The terminology chosen here is intuitively based; like the terms connected and locally connected used in topology where neither one implies the other. They are chosen because they address the same issue and one is “local” in nature.

At each stage of the following algorithm elements of a locally strongly equivalent subset will be replaced by a single attribute from the subset.

2.5.3.4 Reduct Generation Algorithm

Given: $f_A = f_1 \vee \ldots \vee f_k$ where $\{f_1, \ldots, f_k\}$ is a simple cover of f_A.

Step 1. In each component f_i of the simple cover, apply the absorption law to eliminate all conjuncts χ_D where there exists a conjunct χ_C such that $C \subseteq D$.

Step 2. Replace each locally strongly equivalent subset of attributes in each simple cover component J_i by a single attribute that represents this class. A strongly equivalent subset is identified in each component J_i if the corresponding set of attributes is simultaneously either present or absent in each indexing subset of its conjuncts.

Step 3. In each component J_i of the simple cover select an attribute $a \in A$ which belongs to the largest number of indexing sets C_i, numbering at least two, and apply the expansion law. Note $a \in C_i \Rightarrow \chi_{C_i}(\{a\}) = 1$. Write the resulting form as a disjunction . $f_i = f_{i1} \vee f_{i2}$

Step 4. Repeat steps 1 through 3 until f_A is in simple form.

Step 5. For each component f_i of the resulting simple form substitute all locally strongly equivalent classes for their corresponding attributes; i.e., replace each function χ_C by χ_C where $C= \cup\{[a]_i : a \in C\}$.

Step 6. Calculate the reducts Red(f_i).

Step 7. $\bigcup_{i=1}^{p} \mathrm{Red}(f_i)$ Determine the minimal elements, with respect to the inclusion relation, of the set, where $J_A = J_1 \vee \ldots \vee J_p$. These minimal elements are the elements of Red(A).

2.5.3.5 Example

To illustrate the reduct generation algorithm consider the discernibility function (without the explicit χ notation)

$$f_A = \{a, b, c, f\} \wedge \{b, d\} \wedge \{a, d, e, f\} \wedge \{b, c, d\} \wedge \{b, d, e\} \wedge \{d, e\} \quad \textbf{(2-26)}$$

1. Since $\{b, d\} \subset \{b, d, e\}$ and $\{b, d\} \subset \{b, c, d\}$ the absorption law is used to eliminate conjuncts 4 and 5 and get an equivalent discernibility function:

$$f_A = \{a, b, c, f\} \wedge \{b, d\} \wedge \{a, d, e, f\} \wedge \{d, e\} \quad \textbf{(2-27)}$$

2. $\{a, f\}$ is a strongly equivalent class so we can represent it by a single attribute *g* which yields:

$$f_A = \{g, b, c\} \wedge \{b, d\} \wedge \{g, d, e\} \wedge \{d, e\} \quad \textbf{(2-28)}$$

3. In the remaining function, attribute d is the most frequent so apply the expansion law with respect to this attribute to obtain:

$$\begin{aligned} f_A &= f_1 \vee f_2 \\ &= \{d\ \} \wedge \{g, b, c\} \vee \{g, b, c\} \wedge \{b\} \wedge \{g, e\} \wedge \{e\} \\ &= \{d\ \} \wedge \{g, b, c\} \wedge \{b\} \vee \{e\} \end{aligned} \qquad \textbf{(2-29)}$$

where the simplification in the last step resulted from the absorption law.

4. All functions f_i are in simple form.

5. Substituting all strongly equivalent classes for their equivalent attributes:

$$\begin{aligned} f_A &= f_1 \vee f_2 \\ &= \{\ \ \} \wedge \{a, f, b, c\} \vee \{b\} \wedge \{e\} \end{aligned} \qquad \textbf{(2-30)}$$

6. Reducts which correspond to the simple cover functions are

$\mathrm{Red}(f_1)=\{\{a,d\},\{d,f\},\{b,d\},\{c,d\}\}$

$\mathrm{Red}(f_2)=\{\{b,e\}\}$

7. The reducts of *A* are obtained by determining the minimal elements of the set

$$\bigcup_{i=1}^{2} \mathrm{Red}(f_i) = \{\{a, d\}\{d, f\}\{b, d\}\{c, d\}\{b, e\}\} \qquad \textbf{(2-31)}$$

from which it can be concluded $\mathrm{Red}(A) = \{\{a, d\}\{d, f\}\{b, d\}\{c, d\}\{b, e\}\}$. (The reducts of *A* are obtained by “throwing away” supersets in $\bigcup_{i=1}^{p} \mathrm{Red}(f_i)$; in this example there are no supersets.)

2.5.3.6 Results of Computer Simulation

Simulations were run using MATLAB 5.2 on test data generated randomly. A random number generator provided uniformly distributed numbers to represent each attribute of each record. These values were multiplied by eight and then the fractional part was truncated. This resulted in integer attribute values between zero and eight. The number of attributes varied from 10 to 40 in steps of 5 and the number of records varied from 10 to 40 in steps of five. All simulations were accomplished using a dual Pentium Pro 200 MHz computer using 256MB of memory. Figure 2-2 illustrates how the run times increase with problem size using the Expansion Algorithm. Note the abscissa is $\log_{10}$ of the run time. The curves shown are for 10, 15, 20, 25, 30, 35, and 40 attributes. Note that the computational time is growing exponentially.

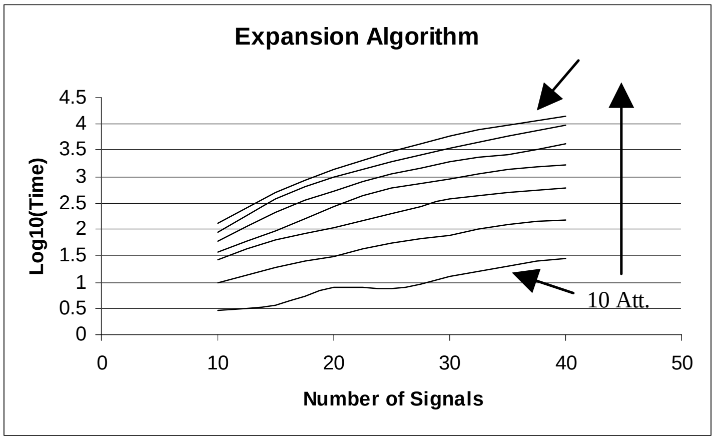


Figure 2-2 Expansion Algorithm Run Times

Figure 2-3 shows the difference in time to run the problems using the Elimination Method and the new Distribution Algorithm. The graph only shows the results for 10 and 15 attributes. This is because when the problem size was larger than this, the elimination method required so much time that results could not be obtained without the simulation running for many days! Note that the time expressed is the $\log_{10}$ of the time.

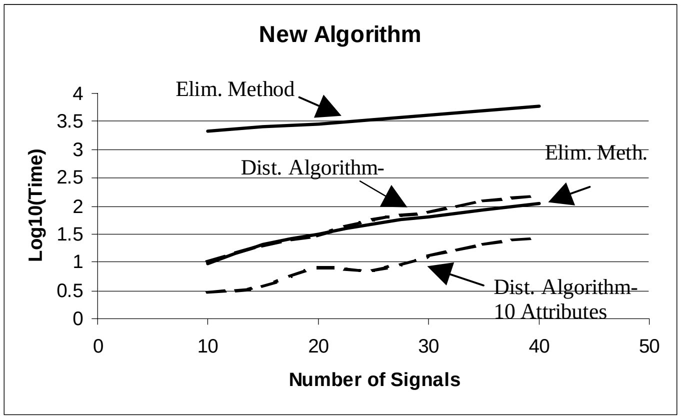


Figure 2-3 Time Savings of the Distribution Algorithm vs. the Elimination Method

Figure 2-4 shows the run times for the Distribution Algorithm with and without strong equivalence. Incorporating strong equivalence into the Expansion Algorithm does cost computational time. However, as seen in Figure 2-4, the time savings can be significant (as much as 50%) when strong equivalence is present.

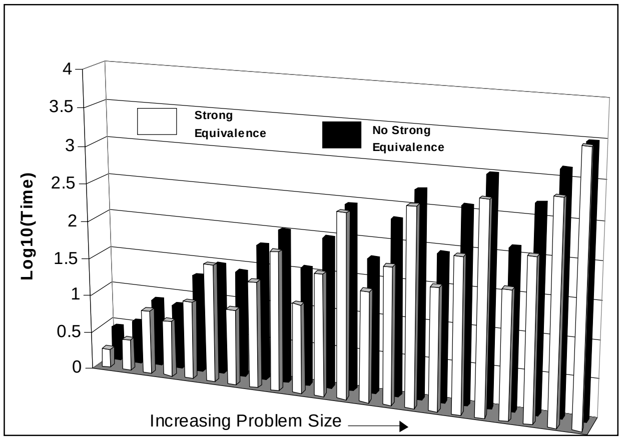


Figure 2-4 Time Savings When Strong Equivalence is Present vs. When it is Not

2.5.4 Summary

All the methods presented in this chapter provide a means to calculate reducts. Skrowron [40] has shown that the process of calculating all reducts is NP-hard. That is, the time required grows non-polynomially as the number of attributes increases. Researchers have found that this limits rough set theory to problems with few attributes. Even algorithms based on the discernibility matrix or the expansion method, though saving time, still have not solved the computation time problem. A more efficient method of finding near minimum (minimal) reducts will be introduced in Section 6.4.5

3 Feature Selection and Fusion

The performance of a classifier has an interdependent relationship between sample sizes, number of features, and classifier complexity. If a simplistic table look-up technique is employed, where the feature space is divided into labeled classes, then an exponential function of the number of features is required for the training set [2]. Since in real world problems, especially in the area of military target recognition the amount of data is limited, it is important to limit the number of features required by the classifier. This is the first reason to keep the number of features small. The second reason is that a limited yet salient feature set simplifies the classifiers [15]. One must be careful that a reduction in the number of features does not lead to a loss in discrimination power and reduced accuracy of the classifier. It should be noted that rough set theory assures that this does not happen. This chapter will review three methods of feature generation; principal component analysis, auto-associative neural networks, and the self organizing map. Three methods of feature selection; exhaustive search, branch and bound search, and sequential forward selection are also summarized.

One area of great interest to the military is the area of information fusion. Information from disparate sources, even of low quality, when combined can yield a synergistic effect and provide very good results. This method of combining results has been used by many researchers to improve classifier performance. Mitchell [22] in his StaF classifier used fusion of information from observations within a signal and then fused this with results from five consecutive observations to improve performance of his classifier. One of the more popular methods in machine intelligence and expert systems is the Dempster-Shafer approach which will be summarized. Since rough set theory generates many classifiers a means of combining the results must be developed or determined and will be described in Section 6.6.

3.1 Feature Generation

Feature generation, or as it is sometimes referred to in the literature feature extraction, differs from the idea of feature selection, although these terms are frequently used interchangeably. **Feature selection** is a process wherein the best subset of features is selected from the input feature set. **Feature extraction** refers to methods that create new features through the use of transformations or some combination of the original feature set and then selects the best from this new set. In most feature generation methods a new dimensional space, of dimensions less than the original space is determined. Thus feature extraction and dimensionality reduction are achieved. Several of the more popular methods of feature generation are covered in the following sections. The research in this dissertation uses wavelets to generate a richer feature space of higher dimensionality relying on multi-class entropy (Section 4.5) and rough set theory (Section 2.5) to perform salient feature selection.

3.1.1 Principal Component Analysis

Principal component analysis (PCA) is probably the best known orthogonal transform based feature generator. This is also known as the **Karhunen-Loeve transform**. PCA transforms a set of n dimensions into another set of $d \leq n$ uncorrelated dimensions while maintaining as much of the variance as possible. PCA produces the eigenvalues and eigenvectors of the system under study. In reducing dimensionality only the eigenvectors associated with the largest eigenvalues are retained. The process of PCA is computationally intensive and is often not feasible for real time applications. PCA is useful for data compression, feature generation, and classification. Since PCA reduces the dimensionality of the data, classifiers based on PCA can be simple and fast.

Mathematically PCA involves solving the equation $RW = W\Lambda$ where R is any real square matrix, W is the eigenvector matrix, and Λ is the diagonal eigenvalue matrix. For signal processing applications R is the full covariance matrix of a zero-mean stationary random signal [35]. Thus, PCA is a linear method which projects data into a set of eigenvectors with scale proportional to the eigenvector's magnitude.

One of the problems with PCA is that it is sensitive to "outliers" [28] or data that appears to be far from the mean. This problem is often addressed by removing these samples from the data. In the case of HRR data it is difficult to determine if such a signal is corrupt or is a valid training sample. The process of elimination may not be straightforward.

Techniques for performing PCA include singular value decomposition (SVD) [33], Hotelling's power method [13], and Hebbian techniques such as Oja's rule [9]. These analytical techniques are computationally intensive and may not be feasible for real time applications. Further these techniques rely on basic assumptions regarding data, such as a zero mean stationary random process which may or may not be true. PCA should be primarily limited to Gaussian distributions. The resulting eigenvectors while orthogonal and uncorrelated are not necessarily selected to optimize classifier performance.

3.1.2 Auto-associative Neural Networks

Neural networks have achieved a lot of press as to their capabilities. Dr. Steve Gustafson of the Air Force Institute of Technology has stated that "neural networks are just statistics for the uninitiated." Regardless, they have found their way into the tool box of researchers [6]. In order to be used as a feature generator and dimensionality reducer a special neural network called an **auto-associative neural network** is used. Figure 3-1 shows an example of an auto-associative neural network. In this example the same training signals are applied to the input and output side. Training proceeds as for a typical neural network using backward error propagation. If the node transfer functions are linear, then the network will find the first three principal components as in PCA analysis. The addition of two hidden layers using sigmoidal transfer functions will yield a nonlinear subspace mapping [15]. The amount of nonlinearity is limited to the size of the middle layer. The illustrated network will reduce the dimensionality of the problem from five dimensions to three dimensions.

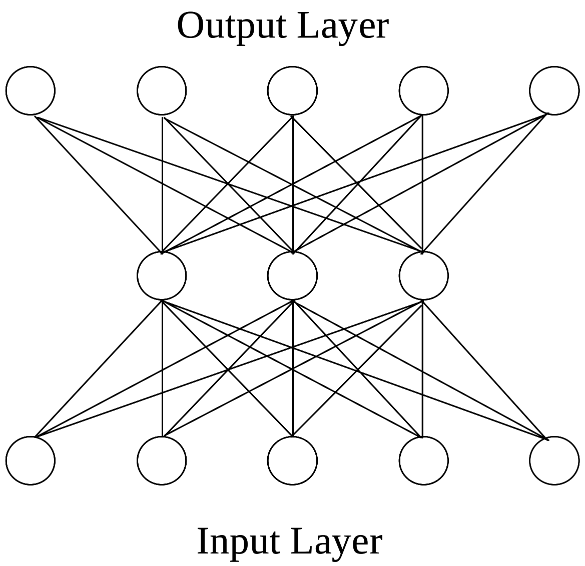


Figure 3-1 Auto-associative Neural Network

3.1.3 Self-Organizing Map

Another method of feature generation is the **self organizing map** also known as the **Khonen Map** [17]. This method can be used for nonlinear feature extraction. This type of network is interesting in that it will learn to detect regularities and correlations in the inputs and adapt its response accordingly. A further feature of this network is that classes that are physically close together in the input space are also close together in the network.

This network is normally arranged in a one, two, or three dimensional grid. Every neuron is connected to all the inputs. Therefore the weights on the connections for the neurons form a d dimensional weight vector. The input signals are presented to the network in random order during the training phase. The first step is to identify the neuron whose vector is closest to the input vector. Once this is accomplished then all the neurons in the neighborhood (which is defined by the structure of the network grid) of that vector have their weights changed to move their vectors closer to the training vector. The farther away from the "winning" neuron the less the weights are adjusted. This means that the weight vectors of neighboring nodes would have been close in the original input space. Effectively a Kohonen network represents an unsupervised nonlinear mapping from the original input space to the feature space of neurons with dimensionality which corresponds to the dimensionality of the neuron grid. One problem with this approach is that the optimum grid dimensionality and the number of neurons are unknown. This may be a serious drawback if the dimensionality of the problem feature space is large and the number of features is undetermined. ATR is this type of problem. Self organizing maps are similar to k-means clustering where during training only a single neuron weight is modified and the grid dimension is the same as the input dimension.

3.1.4 Summary of Feature Generation

The methods presented in this section are both linear and nonlinear. These methods seek to produce dimensionality reduction and thus a more concise method of describing the training set. They result in transforming the input space to a set of orthogonal vectors with little or no cross correlation. What these methods lack is that the final basis vector set has not been optimized to do classification nor to maintain the information in the training set. Often to work well outliers are removed from the training set. The problem is that they may be valid data points and not “outliers.” This may result in the loss of information. It is entirely possible that the features generated while being orthogonal are not the best for the classification problem. Feature generation using wavelets suffers from none of these problems while showing a great benefit as discussed in Chapter 5.

3.2 Feature Selection

Feature selection involves a process where features, either generated or natural, are selected such that classification using this subset results in the smallest classification error. Feature selection becomes more important as the number of features increase. This increase may come from increased resolution of sensors or the fusion of multiple sensors.

3.2.1 Exhaustive Search

The obvious simplest approach to feature selection is to try all possible combinations of features of a selected size. Based upon some criterion, such as classifier performance on the test set, the subset with the best performance would be selected for the classifier. This approach is combinatorically explosive where for even modest numbers of attributes and small subsets of attributes computations become prohibitive. Cover and Van Capenhout [4] related that in order to guarantee the optimality of a 12 feature subset selected out of 24 possible features would require approximately 2.7 million possible subsets to be evaluated. The only optimal (based on monotonic criterion functions) that avoid exhaustive search are methods based on the branch and bound algorithm.

3.2.2 Branch and Bound Search

The artificial intelligence community has long been struggling with how to solve NP-hard problems. A typical approach that has been used on this kind of problem is the branch and bound search [36]. One problem of the NP-complete genre is the traveling salesman problem.

The Traveling Salesman Problem

> A salesman has a list of cities, each of which he must visit exactly once. There are direct roads between each pair of cities on the list. Find the route the salesman should follow so that he travels the shortest possible distance on a round trip, starting at any one of the cities and then returning there.

Using exhaustive search as in the previous section the time required to perform this search is O(N!). A better strategy is to use a branch and bound search. In this method paths are generated and the shortest path is recorded. As soon as a path exceeds this length, it is abandoned. This guarantees that the shortest path will be found. Although more efficient than exhaustive search it is still exponential in time complexity but can be used for solving larger problems. In general, branch and bound search is guided in the process of finding the best set of features by determining bounds on the final criterion value at each intermediate stage. This method requires the use of a criterion function which is monotonic. This means that the performance of a feature subset should improve when a feature is added to it. Most criterion functions do not satisfy this property since they are not monotonic[15] [12]. The main advantage of the branch and bound technique is that search paths that are worse than others can be abandoned early. A portion of this technique will be used in the reduct determination of section 6.4.5.

3.2.3 Sequential Forward Selection

One of the non-optimal methods for feature selection is the sequential forward search. In this method, starting from the initial state, the best single feature is selected and then one feature at a time is added which in combination with the other selected features maximizes the criterion function. This process continues until the goal state is reached [36][15]. The main attraction to this method is that it is computationally fast. The main drawback is that once a feature is added to the subset it cannot be removed. Once the goal state is reached a branch and bound procedure can be used to find all other paths to the goal stopping each subsequent search when it exceeds the shortest path found so far.

3.3 Fusion - Dempster-Shafer Method

In the area of fusion of information, the theory of evidence [38] developed by Dempster and Shafer (D-S) is one of the earliest and still best used methods today. D-S is a generalization of Bayesian theory. D-S assumes that the hypotheses it deals with are singleton, mutually exclusive, and there is no level of uncertainty assigned to the decision. D-S uses singleton hypotheses regarding a system being examined and associates a belief with each one. D-S provides a mechanism where the beliefs associated with a hypothesis can be combined to make a statement regarding the overall belief. This is the method used by Mitchell in the StaF classifier [23] to combine beliefs regarding features to yield a classification and a confidence associated with that classification. Mitchell went to great lengths to establish the probability density functions associated with the feature position and magnitude. Since these probabilities are not known they can only be estimated resulting in a potential source of error.

Rough set theory will yield a multiple number of classifiers based on the number of reducts found. It is well known that by combining the advice of many advisors one can reach a better decision. Therefore it seems logical that if a way can be devised to combine the output of many classifiers the result should be better. The D-S method does not fit well for this application. The main problem is estimating the PDFs and calculating the probabilities associated with each classification. Therefore, in Section 6.6 a new method for fusing the results of the classifiers for a given partition and the classifiers for all the partitions is introduced.

3.4 Summary

Feature selection is one of the most important steps in the construction of a classifier. In rough set theory the process of feature selection is called reduct generation. In Section 2.5.1 the process of finding a reduct is the same as discussed for exhaustive search. All possible combinations are tried. The faster reduct finding method as discussed in Section 6.4.5 more closely resembles a combination of sequential forward feature selection and a branch and bound search. The criterion function is not based on classification ability but on information preservation (reducing the number of ambiguities).

Fusion is important to achieve robustness in the RST classifier. Many partitions and many reducts must be combined to achieve high declaration rates and a high probability of correct classification. The Dempster-Shafer approach has been successfully used by other researchers. The primary problem is that D-S requires that probability density functions be approximated. This is difficult due to a lack of data and the fact that the shape of distribution itself is unknown. Rather than make arbitrary assumptions a radically different and new method is presented in Section 6.6.

4 Data Preprocessing

As in any application, the proper preprocessing of the data is essential to solving a problem. Data is normally preprocessed to remove noise, enhance, and equalize the signals. This chapter will present the procedures used for this effort.

4.1 Data

The data set used in this research consists of synthetic HRR returns on six targets. For each target there are 1071 range profiles consisting of 128 range bins. The value of each range bin is an integer between 0 and 255. The pose of the target is head-on with an azimuth range of $\pm 25^{o}$ and elevations of -20 ^{o} to 0 ^{o} in 1^{o} increments as illustrated in Figure 4-1.

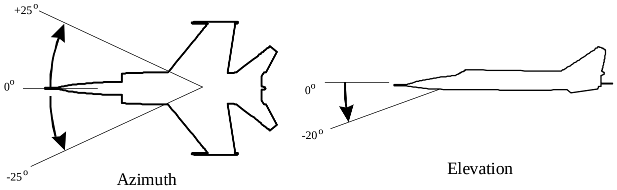


Figure 4-1 Target Viewing Aspect

This data is divided into two sets, one for training and the other one for testing. The training set consists of 25% of the data, randomly selected, and the test set 75% of the data (the remaining data). The small training set permits faster training, facilitating algorithm development, and debugging. The training set was constructed by using a random number generator to select 25% of the azimuth and elevation angles and then by selecting signals from each target class with these values. All remaining signals were placed into the test set.

4.2 Partitioning - Focused Reducts

One of the problems with rough set classification is that the determination of all the reducts (as described in Chapter 2) is a NP-hard problem [40]. Even using methods described in [44] the HRR ATR problem is much too large. Therefore, it is necessary to only use a small subset of the available range bins. It was hypothesized that by careful selection this subselection may also be advantageous to the classification process as the smaller data sets would force the rough set classifier to focus on either local or more global features depending on how the data is partitioned. The range bins used were selected in a number of different ways. A signal was divided into partitions consisting of all the data, one-half of the data, one-quarter of the data, and one-eighth of the data. There were two ways of selecting data from each partition size. On the partition using one-half of the data, the first selection consisted of two sets, the first 64 range bins, and the last 64 range bins, Figure 4-2. The second selection consisted of two sets, the even numbered range bins, and the odd numbered range bins, Figure 4-3. For the partitions where the data is in fourths, range bins 1-32, 33-64, 65-96, 97-128 were selected. The second selection consisted of range bins {1, 5, 9, 13, …, 125}, {2, 6, 10, 14, …, 126}, {3, 7, 11, 15, …, 127}, {4, 8, 12, 16, …, 128}. Similar procedures were used for the other partition sizes. These two types of partitions are called block partitioning and interleave partitioning. Other partitions and selection schemes are possible although were not considered for this research.

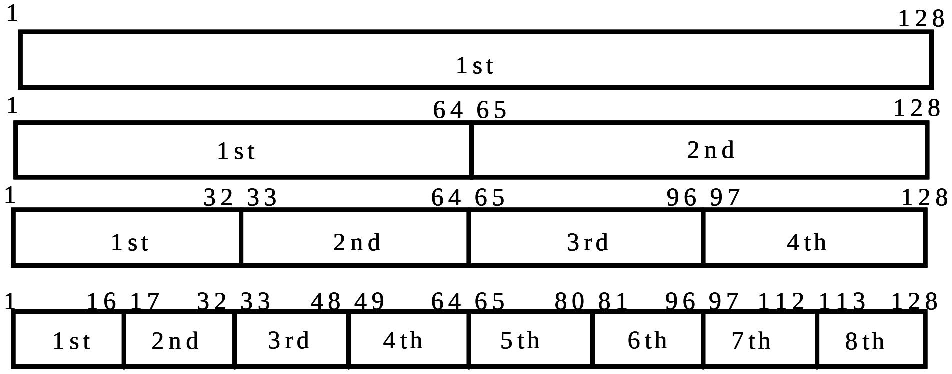


Figure 4-2 Block Partitioning

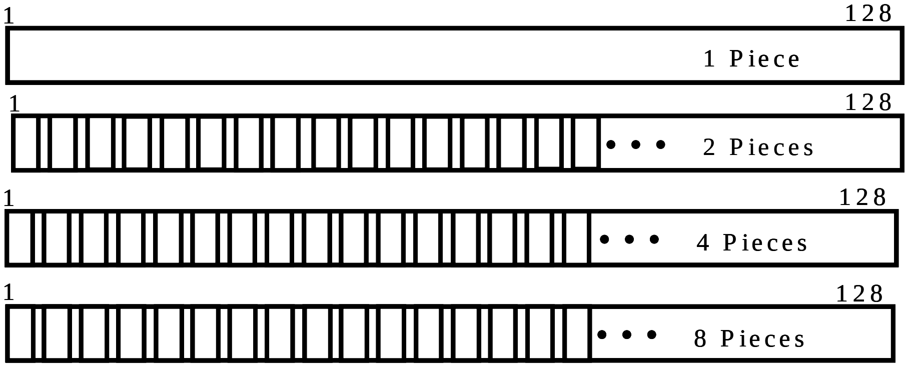


Figure 4-3 Interleave Partitioning

4.3 Normalization

Radar signals can have a lot of variability due to the various radar parameters and other factors exogenous to the ATR system. In an attempt to remove these effects and focus on the relative signal strengths in the range bins the signal is normalized. The signal is normalized following the partitioning described in Section 4.2. This is accomplished by dividing each signal value by the L2 norm across the signal's range bins. The **L2 norm** is defined as:

$$N = \left(\sum_i |x_i|^2\right)^{\frac{1}{2}} \qquad \textbf{(4-1)}$$

Normalization facilitates numerical analysis of the results and effectively reduces noise introduced by round off errors. It also facilitates hardware implementation by keeping signal values within specified machine precision.

4.4 Quantification

Rough sets are different than fuzzy sets. Where fuzzy sets may be characterized as being concerned with how gray a pixel is rough sets are concerned with how large a pixel is [3]. This concept of size translates into labeling for the HRR problem. Therefore, we must choose a scheme to label the data. The greater the number of labels the finer the division of the classification space and presumably better performance.The process of labeling range bin value ranges is known as quantification. The normalized values in the range bins represent an infinite quantification in that every label is drawn from the real numbers. Using this original labeling each signal is unique and belongs to its own equivalence class. What is desired is a labeling where targets of the same class have the same signal after labeling. It is highly unlikely that this is possible. To determine an optimum labeling one would have to try placing dividing points between each division of data in each range bin. A classifier would need to be developed and tested against each possible labeling scheme. Obviously for any problem representative of the real world this would take much too long. Skowron [39] showed that this problem of quantizing real valued attributes is NP-hard. Handling real valued attributes has vexed rough set researchers for a long time. As stated earlier, rough set theory is more concerned with the size of a pixel rather than how gray it is. This concept of “size of a pixel” is the quantification problem. The finer the resolution the better job of defining the boundaries of the target classes will be. However, as the class boundaries become of finer resolution, more training data points must be used to define the boundary. In many problems this quantity of data may not be available.

4.5 Binary Multi-Class Entropy

For this effort a method of binary labeling based on **multi-class information entropy** was chosen.

Information entropy is a concept introduced by Shannon [50]. He considered a single random variable taking n values with probabilities $p_1,...,p_n$ and defines its entropy as:

$$H= \sum_{i=1}^{n} p_i \log(p_i) \qquad \textbf{(4-2)}$$

This concept of entropy will be expanded to multi-class entropy to define a point with the range of values for a single range bin which will be the point of maximum information for discriminating among the various target classes.

Now assume a range bin (attribute) across all training signals is defined as:

$$a_i = \{x_1, \ldots, x_n\} \qquad \textbf{(4-3)}$$

Let

$$a_t \equiv \{x_j > x_t | (x_j \in a_i)\} \qquad \textbf{(4-4)}$$

Rather than forcing an assumed distribution on the data an approximation can be used to obtain the probabilities. Thus the probability that a given attribute satisfies the threshold x_t can be estimated at each point x_t as the quotient of cardinalities:

$$P_t = \frac{|a_t|}{|a_i|} \qquad \textbf{(4-5)}$$

For simplicity of notation for each threshold value x_t, P_t will be represented as P_1 and its complement $P_0 = 1\text{-}P_1$ will represent the probability that a given attribute does not satisfy the threshold. Likewise for each class the probabilities that a given attribute of a selected class satisfies a threshold can be estimated as:

$$P_{tc} = \frac{|a_{tc}|}{|a_i|} \qquad \textbf{(4-6)}$$

where $a_{tc} = \{x_j > x_t | x_j \in a_i, j \in c\}$ and c represents a set of signals belonging to class C. Finally class probabilities are estimated from:

$$P_c = \frac{|a_c|}{|a_i|} \qquad \textbf{(4-7)}$$

where $a_c = \{x_c | j \in c\}$.

Each column (range bin) of the training set is searched sequentially to establish the optimum point which best separates signals of the various training classes. The quality of the range bin partition is measured by the entropy-based information index defined as:

$$I = 1 - \frac{\Delta E}{E_{max}} \qquad \textbf{(4-8)}$$

where relative entropy ΔE is defined as:

$$\Delta E = -\sum_{t=0}^{1} \sum_{c=1}^{n_c} p_{tc} \log(p_{tc}) + \sum_{t=0}^{1} p_t \log(p_t) \qquad \textbf{(4-9)}$$

and

$$E_{max} = -\sum_{c=1}^{n_c} p_c \log(p_c) \qquad \textbf{(4-10)}$$

where n_c is the number of classes in the training set, p_c, p_t, p_{tc} are the probabilities of each class, thresholded attribute probabilities, and joint probabilities respectively.

The point at which the relative entropy is minimum for the range bin is the division point that will provide the maximum amount of information for separating all the target classes. It would be informative to visually determine if the entropy truly provides a good labeling division point. Further, it would be informative to determine if higher entropy index values are associated with range bins which do a better job of separating classes. Figure 4-4 illustrates how entropy works at defining a division point. In this figure one target class is clearly separable from the rest. The vertical line represents the best division point for labeling. It appears to be at the point where one would visually put it. This point will be used as the labeling break because it is optimum for separating <u>all six</u> target classes. Figure 4-5 shows another range bin. This one is not as clear but the division point appears to make sense. Figure 4-6 shows a range bin where there is no discernible point at which it makes sense to make a division. Also shown with each figure is the entropy index for that range bin. The greater the index value the more useful the range bin for the classification process. Figure 4-4 is by far the best choice while Figure 4-6 is so low that it makes no sense to select this range bin. Note that the highest entropy index is two orders of magnitude better than the low entropy index. These figures clearly show that multi-class information entropy provides a good theoretical division point for labeling the range bins as well as the measure of quality of the range bin.

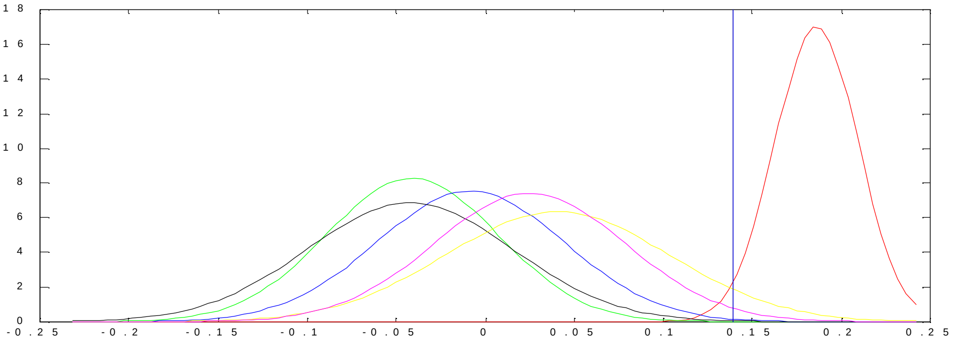


Figure 4-4 High Entropy (Entropy Index =.138810)

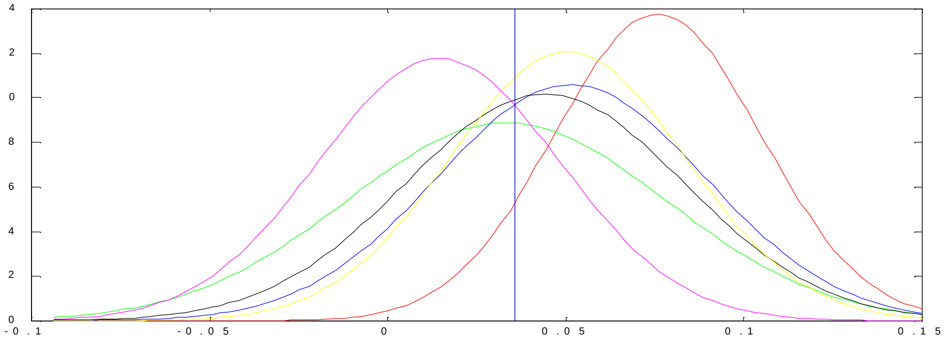


Figure 4-5 Medium Entropy (Entropy Index =.035594)

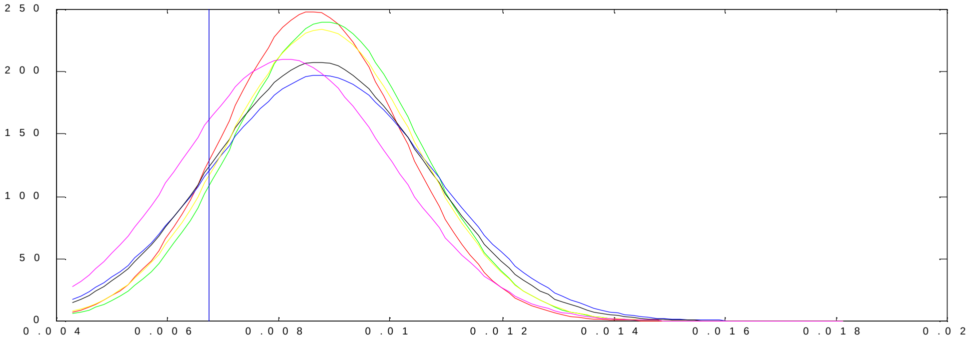


Figure 4-6 Low Entropy (Entropy Index =.006740)

4.6 Fuzzification

In dealing with measured data there are problems associated with noise introduced by the measurement equipment itself, variability among different pieces of the same equipment, round-off error, and the physics of the problem. When developing ATR systems, one cannot expect that the data upon which the system is based will match exactly the measurement taken by operational systems. Therefore, most ATR systems allow for some variability through the concept of fuzzification. Fuzzy systems account for the uncertainty associated with the real world by providing a way to quantify the set membership function. In other words it provides a means to express the degree of uncertainty of the classification process.

4.6.1 Relation Between Fuzzy Classification and Rough Set Theory

Vagueness is represented in fuzzy sets by membership functions which map the universe to a unit interval containing membership values [31]. This idea is shown graphically in Figure 4-7. Each class represented has an overlap region where class membership is less than one. Values along the x-axis may belong to a given class or they may belong only partially to that class. The degree of membership may vary between 0 and 1. In RST the x-axis would represent values of an attribute. This research proposes only two labellings for an attribute 1 or 0. If fuzziness were to be used, a membership function would be set-up around the division point and would represent how 1 or 0 the labeled attribute was, based upon the test set original attribute value. This, however, does not give sufficient information to determine the target class of the test signal. It must be combined with other attributes of the reduct to determine the class of that signal according to its partition. Then the partitions are fused to yield a classification. A complete explanation of this process may be found in Chapter 6.

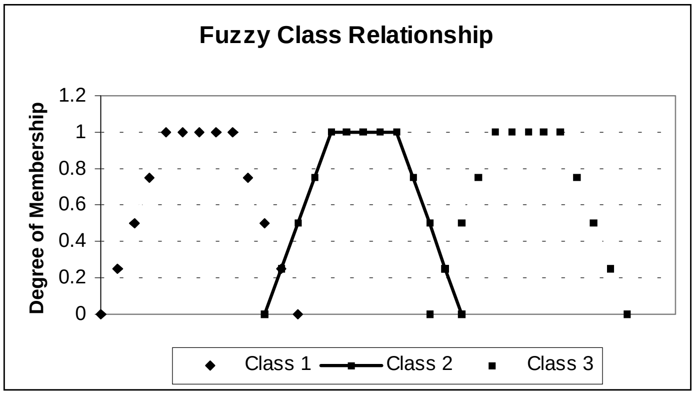


Figure 4-7 Fuzzy Relationship

The normal concept of fuzzy sets would not work well for this version of a rough set classifier. Therefore another way to introduce imprecision, vagueness, and noise into the classification needed to be developed.

The training set is assumed to be pristine in nature and therefore the information entropy determined division point is assumed to be crisp. However, test set signals are labeled somewhat differently than the training signals. The division point for each range bin in the test signal must be the same as for that same range bin in the training signal. Using this method yields a sharp labeling of the data. Values close to this division point could possibly be "mislabeled" due to noise or some other reason. Therefore a provision was made in developing the classifier to provide a buffer zone around the division point. This buffer zone is defined:

$$d = b * \min(|x_d - \min(x_i)|, |\max(x_i) - x_d|) \qquad \textbf{(4-11)}$$

where d is the distance from the division point, b (**fuzz factor**) is the portion of the smallest distance to be used, x_d is the division point, and x_i is the range bin value. The **buffer zone** is then defined as the distance $x_d \pm d$. Any value in the buffer zone is treated as "don't care"; i.e., that range bin will not be considered in the classification process for that signal. Notice that this labeling (with do not care) is not a typical use of the membership function and indicates uncertainty in the rough set boundary. Thus "roughness" of the rough set boundary is explicitly exploited to improve recognition accuracy. The effect of this “fuzz factor” on the classification performance is shown in Figure 7-7.

4.6.2 Bar Graph of Distance Values

In order to better understand how the fuzz factor affects the performance of the classifier histograms were generated from the test data for each of the various data partitions and are presented in Figure 4-8 through Figure 4-36. The y-axis is the frequency count. The x-axis represents the ratio of the distance from the labeling point to the actual value divided by minimum of the distance from the labeling point to the maximum bin value or the minimum value from the training set; i.e.,

$$d_x = \frac{|x_i - x_d|}{\min(|x_d - \min(x_{it})|, |\max(x_{it}) - x_d|)} \qquad \textbf{(4-12)}$$

where d_x is the bar-graphed distance, x_d is the actual value of the range bin from the test set, and x_{it} are the values of the range bin from the training set. These distance values are calculated and histogrammed for each range bin in the reduct and each signal in the test set. The way to interpret the histograms is that if the fuzz factor was 0.05 then the number of values (range bins in all the signals) as represented by the first bar would be marked as “don’t car.” If the fuzz factor was 0.10 then the number of values indicated by the first two bars would be marked as “don’t care.”

In reviewing these histograms it is apparent that there are many cases where most values are in the first bin such as in Figure 4-13. What this means is that when the fuzz factor is 0.05 then those range bins will not be considered in the classification process for that partition. Partitions where this is most evident, Figure 4-13, Figure 4-16, Figure 4-21, Figure 4-22, Figure 4-27, and Figure 4-28 are also found (Table 7-2) to have poor classification performance. These figures represent block partitioning at the beginning and end of the signal which is known to contain mostly noise. Therefore, it not surprising to find that the test values fall close to the labeling point as these range bins are likely to have a small dynamic range. In many of the partitions, especially the better performing ones, the distance values are more distributed. This indicates that there is more likely to be more information in that partition.

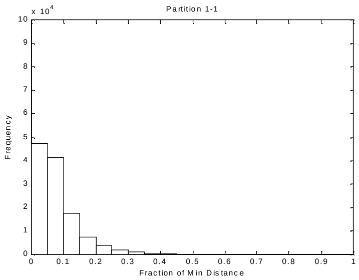


Figure 4-8 Histogram Partition Tst1-1

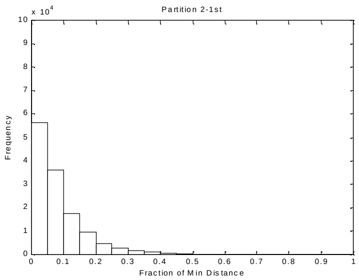


Figure 4-9 Histogram Partition Tst2-1st

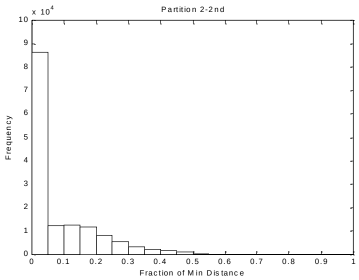


Figure 4-10 Histogram Partition Tst2-2nd

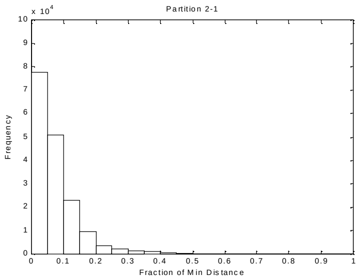


Figure 4-11 Histogram Partition Tst2-1

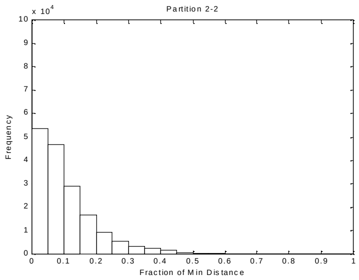


Figure 4-12 Histogram Partition Tst2-2

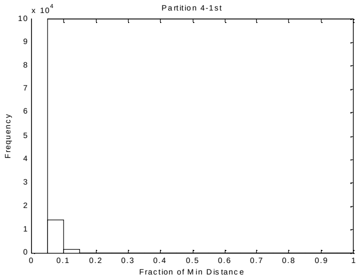


Figure 4-13 Histogram Partition Tst4-1st

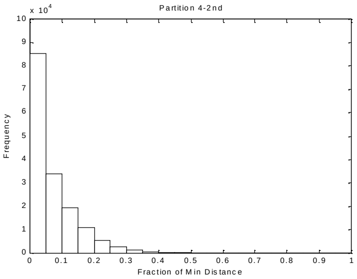


Figure 4-14 Histogram Partition Tst4-2nd

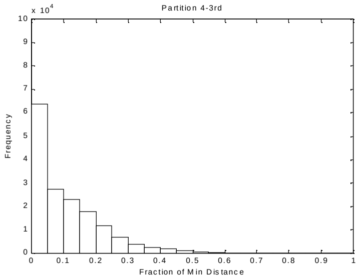


Figure 4-15 Histogram Partition Tst4-3rd

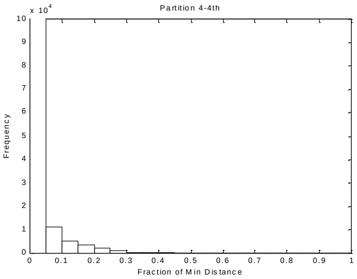


Figure 4-16 Histogram Partition Tst4-4th

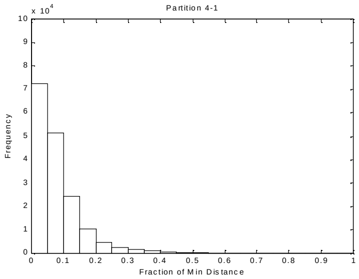


Figure 4-17 Histogram Partition Tst4-1

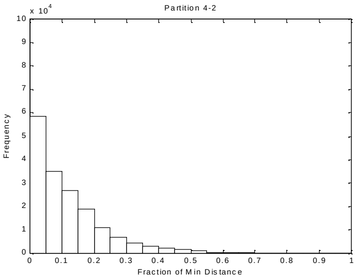


Figure 4-18 Histogram Partition Tst4-2

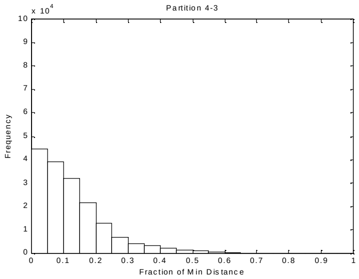


Figure 4-19 Histogram Partition Tst4-3

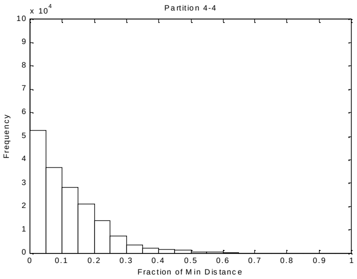


Figure 4-20 Histogram Partition Tst4-4

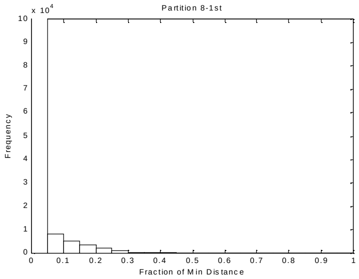


Figure 4-21 Histogram Partition Tst8-1st

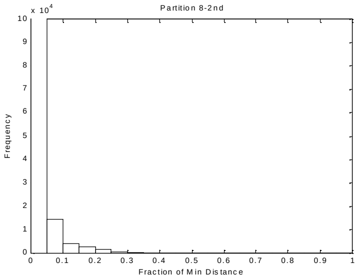


Figure 4-22 Histogram Partition Tst8-2nd

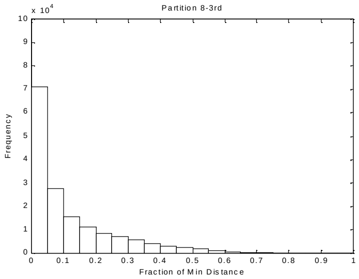


Figure 4-23 Histogram Partition Tst8-3rd

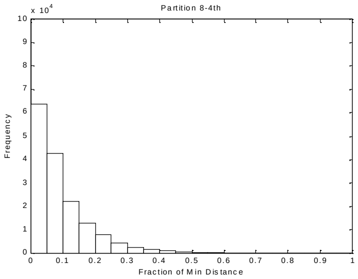


Figure 4-24 Histogram Partition Tst8-4th

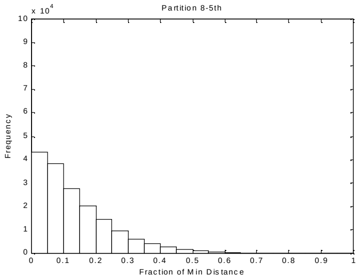


Figure 4-25 Histogram Partition Tst8-5th

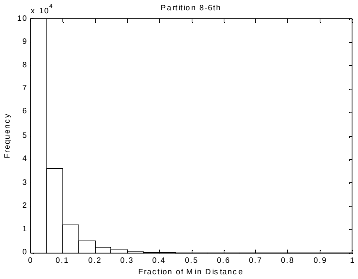


Figure 4-26 Histogram Partition Tst8-6th

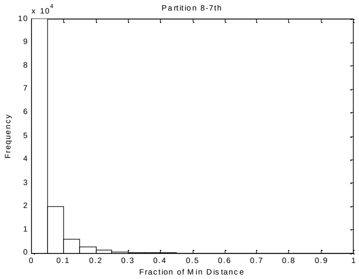


Figure 4-27 Histogram Partition Tst 8-7th

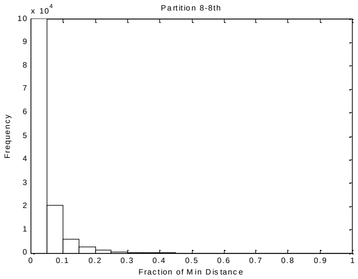


Figure 4-28 Histogram Partition Tst8-8th

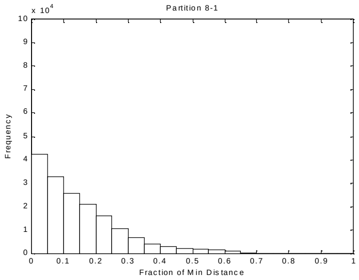


Figure 4-29 Histogram Partition 8-1

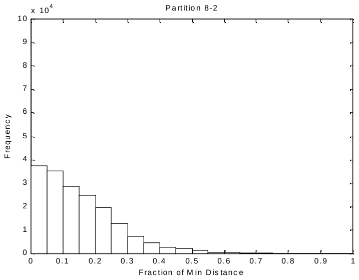


Figure 4-30 Histogram Partition 8-2

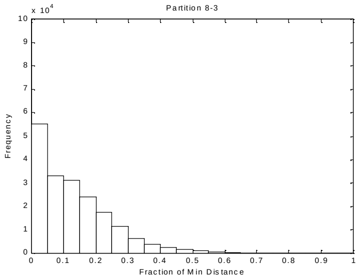


Figure 4-31 Histogram Partition Tst8-3

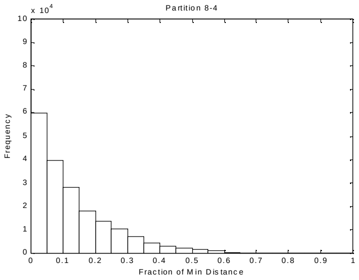


Figure 4-32 Histogram Partition Tst8-4

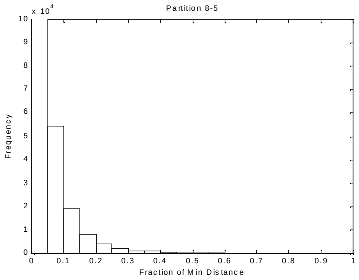


Figure 4-33 Histogram Partition Tst8-5

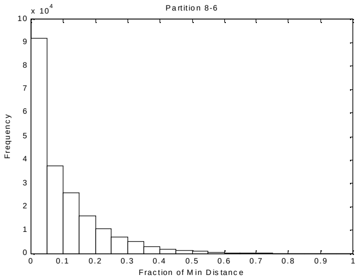


Figure 4-34 Histogram Partition Tst8-6

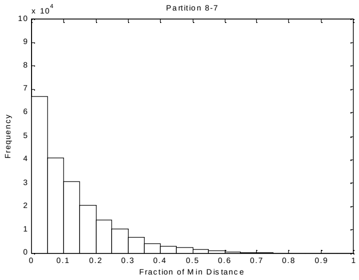


Figure 4-35 Histogram Partition Tst8-7

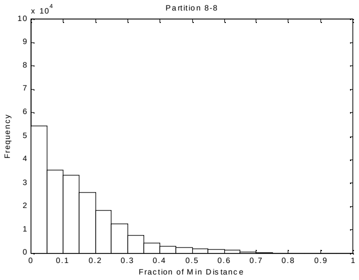


Figure 4-36 Histogram Partition Tst8-8

4.7 Summary

This chapter has covered many areas associated with the data preprocessing steps to ensure that the information in the data is most useful to the classification process. The data was block partitioned in order to force the classifiers to focus on various portions of the signal for local classification power. The interleave partitioning forces more global classification. Signals were normalized to remove some of the effects of the radar and radar parameters and produce a more unified dynamic range. Since rough set theory does not work with real numbers a method was introduced to quantify or label these numbers using a binary scheme. Finally a unique method was introduced to handle uncertainty in the data by creating a “don’t care” region which effectively removes those range bins from consideration in the classification process. Effect of this data preprocessing is revealed in Chapter 7, Classification Results.

5 Wavelets

Most of the work in HRR target recognition has been done by or sponsored by the military. The approaches taken by various researchers as summarized in [1] appear to ignore the benefits that can be gained by proper transformations of the input signal. The **wavelet transform** [2][3][4] is a new tool that has been used in image compression, edge detection, image classification, and more recently in target recognition. When wavelet transforms are used for image compression the most important goal is to minimize the loss of information. In Automatic Target Recognition (ATR) the most important objective is to separate the various target classes [5]. Some researchers have explored the use of wavelets to provide a richer feature space [5][6][7][8]. However, there is little evidence of widespread use of this technique. Mitchell [22] explored the use of a sixth order Daubechies transformation but he limited the analysis to an autoregressive approach to remove low information data from the signature.

Famili [9] found that preprocessing the data allows easier subsequent feature extraction and increased resolution. In the past engineers have used transforms such as the Fourier transform to transform the signal from a time base to a frequency base [10]. Although this is useful for some applications target recognition of HRR signals improved only a little under this transform. The reason for this lies in the fact that the Fourier Transform tells us that a feature occurs somewhere in the signal but not where. Wavelets bring a new tool to HRR signal classification. The benefits of using wavelets [11] are that the new transforms are local; i.e., the event is connected to the time when it occurs. Researchers who have used wavelets for target recognition (especially for HRR) have found that the original feature space can be augmented by the wavelet coefficients and will yield a smaller set of more robust features in the final classifier [7][8][12]. In addition to computational savings [8] investigators have also found that wavelet methods can improve the probability of correct classification (P_{cc}) [6][7]. However, even with improvement in P_{cc} there can be a bias of the wavelets toward one or two classes to the detriment of others [7].

In considering wavelets for ATR serious consideration must be given to the selection of a wavelet family and a wavelet in the family. Lu [13] explored this issue in the context of image coders. In his paper Lu compared two wavelets, one from the Biorthogonal family and the other from the Daubechies family. Using two different metrics Lu observed a slight advantage of the Biorthogonal versus the Daubechies. Stirman using wavelets for ATR explored the use of different wavelets from the Daubechies family and found that the results were similar among the three wavelets [7]. Using the criterion of improving the probability of correct classification it will be demonstrated that there is no statistical advantage of one family (out of four) over any other family thus generalizing Stirman's observation. Any difference in performance that can be observed in a particular application is due to the statistical nature of normal variations in the data.

Other researchers have employed wavelets to assist in HRR target identification [14][15]. Devaney's approach used a sequential decision process where the log likelihood ratios are computed at each scale in the discrete wavelet transform (DWT) and then hypothesis testing is applied at each scale to yield the target identification. Etemad used the multi-scale DWT to reduce the dimensionality of the classification problem. He used the coefficients to build a set of basis functions that yield the largest class separability. These basis functions result in simple and efficient algorithms for classification. The work presented here differs from the efforts of these two researchers. A classifier is employed in this chapter but the focus is not on the classifier but on a method to improve upon the DWT itself. Etemad and Devaney applied the multi-scale DWT one time. This chapter introduces the **iterated wavelet transform**. After the first application of the DWT a subset of coefficients equal to the original number of range bins in the signal is down-selected using the box classifier. A

DWT of this “new signal” is then accomplished. Doing this many times yields a new pseudo wavelet constructed for the problem presented by the training data.

It is not the purpose of this chapter to explore the development of a classifier. However, in order to have a means to test the usefulness of the data transforms a classifier must be used to test the performance and determine which features to select for further transformation. The simple generalized box classifier [16][17][18] has been chosen to evaluate the results. The main objective was to determine which, if any, family of wavelets provided the best feature set for a classifier. A secondary objective was to determine if further wavelet transformations would produce even better classification results. The secondary objective required a method for down selecting features from which to perform further wavelet analysis. In this chapter, using wavelet transformations, it will be shown:

1) wavelets are useful for generating features that improve classifier performance.

2) what family and which wavelet in the family is best.

3) how to mitigate or eliminate wavelet bias towards some target classes.

5.1 Generalized Box Classifier

The classifier used in this chapter is a version of the generalized box classifier [16]. The training set is used to construct the classifier. Each row of the training and test sets is referred to as a signal. The training set S consists of signals having 1024 pseudo range bins.

The first step in constructing the classifier is to sort each column of S from the smallest value to largest value creating a new matrix $\bar{S}$. A matrix $\bar{M}$ is constructed with each element of $\bar{M}$ corresponding to the target type of each element of $\bar{S}$.

The algorithm for constructing the classifier is as follows:

Let i denote the target class, and j the feature number. Set i = j = 1.

Generalized Box Classifier Algorithm

Step 1 Search all columns of $\bar{M}$ to find the column with the largest contiguous cluster of the selected target class *i*. Let $\sigma(j)$ denote the column determined by this procedure σ is(a permutation of the columns of $\bar{S}$). Let $\bar{S}_{n,\,\sigma(j)}$ denote the minimum value in the contiguous cluster and let $\bar{S}_{k,\,\sigma(j)}$ denote the maximum value in the contiguous cluster. The indices n and k correspond to the row indices of $\bar{S}$ with the minimum and maximum values. All signals contained in this cluster are removed from further consideration.

Step 2 Define the jth feature of target class i as the set $a_{ij} = (\bar{S}_{n,\,\sigma(j)}, \bar{S}_{k,\,\sigma(j)})$. Set $i = j + 1$ and repeat this process (go to step 1) until there are no more training signals from target class i.

Step 3 Increment target class i and set j = 1. Repeat this process (go to step 1) until all target classes are accounted for.

The elements a_{ij} are called individual attributes. The attribute set A is defined as the set of all a_{ij}. A transformed signal z is said to be classified as target class i when there exists an attribute a_{ij} such that $z \in a_{ij}$.

The classifier is tested by classifying each of the transformed test signals, z. An n_c x n_c **confusion matrix** C is constructed to represent the results. To construct the confusion matrix first set C = [0]. Each test signal is classified and C is modified as follows. If the test signal i known to be of class j is classified as target type j then $C_{jj} = C_{jj} + 1$. If the test signal i known to be of class j is classified as target type k, then $C_{jk} = C_{jk} + 1$. This process continues until all transformed test signals are classified. In this chapter equal numbers of signals were used to represent each target class for both training and test. Therefore, to obtain the final confusion matrix each element of C is divided by the number of signals for a target class. It should be noted that some test signals may not be classified as any target type. Therefore, it is possible that the rows and columns of the confusion matrix will not sum to one. To evaluate the overall performance of the classifier the probability of correct classification, P_{cc}, is calculated. $\mathbf{P_{cc}}$ is defined for n target classes as:

$$P_{cc} = \frac{1}{n_c} \sum_{i=1}^{n_C} C_{ii} \qquad \textbf{(5-1)}$$

P_{cc} is a random variable value which represents a classifier performance on a given set of data. Therefore, care must be used in utilizing this measure in comparing two different classifiers.

5.2 Wavelet Families

Wavelet transforms have been found useful in a variety of applications. This is because they provide the analyst with an approximation of the signal and a detail of the signal as well. This helps to identify small anomalies that might be useful. For a complete description of wavelet analysis the reader should refer to [10][11]. A brief summary of how the wavelets were used is presented here.

The 1-D **discrete wavelet transform** (DWT) of a signal yields an approximation and a detail of the original signal. Passing the original signal through a low-pass filter then down sampling produces the approximation. Passing the original signal through a high-pass filter then down sampling produces the detail. The corresponding wavelet functions may be obtained by an iterative process involving convolution of the filter coefficients.

Discrete wavelet packet (DWP) analysis begins with the DWT of the original signal. The next level of the DWP analysis calculates the DWT of the resulting approximation and the DWT of the resulting detail. Subsequent levels calculate the DWT of all the approximations and details of the previous level. Since the number of samples of each approximation and detail is approximately half of the number of samples of the input signal to the DWT, DWP analysis must cease when the approximations and details each contain a single sample.

The discrete wavelet packet (DWP) analysis is performed as follows. The length, L, of the normalized signal must be a power of 2. The number of levels of the DWP analysis is $N = \log_2(L)$, beginning with level N, and ending with level 1. At level j the 1-D discrete wavelet transform of a signal of length 2^j is calculated using DWT.m from the MATLAB

Wavelet Toolbox. The outputs of DWT are an approximation and a detail each of length at least 2^{j-1}. The tails of the approximation and detail are all zero-valued, so zero values are trimmed from each tail to obtain lengths of 2^{j-1}. The trimmed approximations and details are used as inputs to the DWT at subsequent levels. The trimmed approximations and details at each level are appended to the normalized signal and placed in the training set. This process is depicted in Figure 5-1. The wavelet functions used in this chapter for the DWT are shown in Table 5-1.

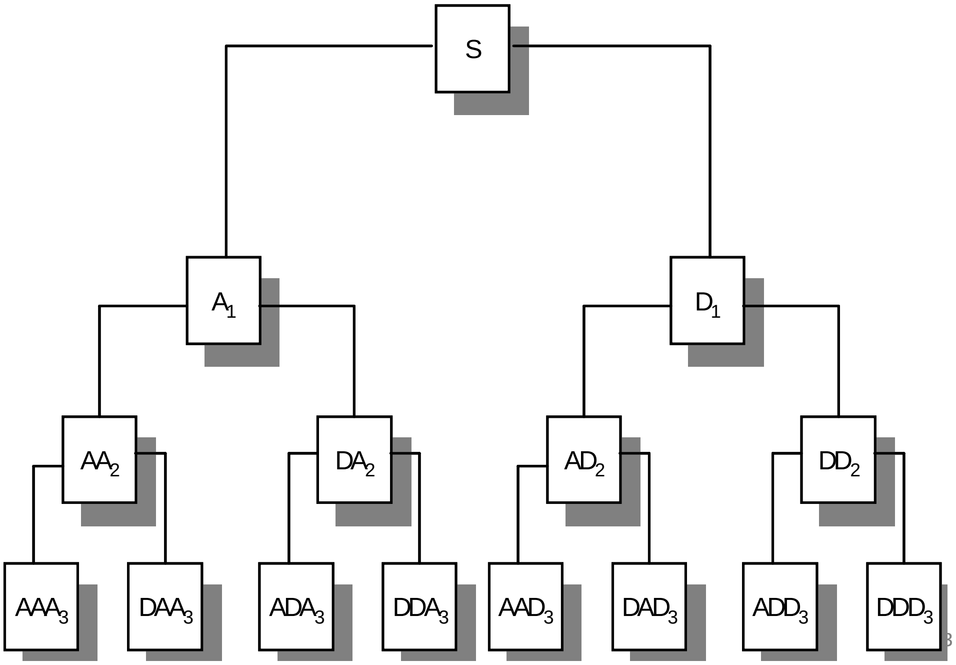
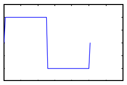
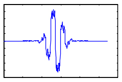
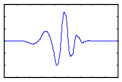
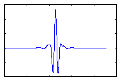
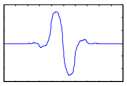
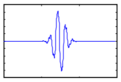
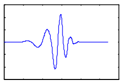
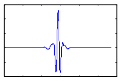
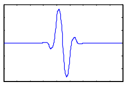
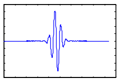
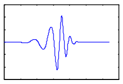
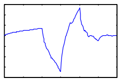
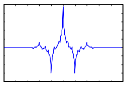
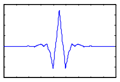
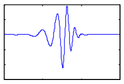
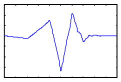
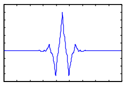
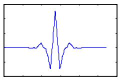
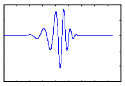
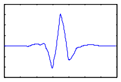
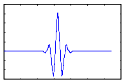
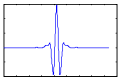
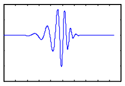
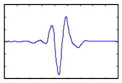
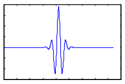
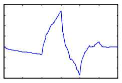
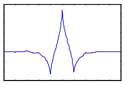
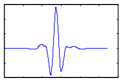
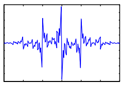
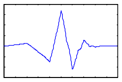
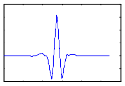
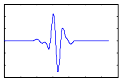
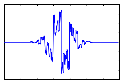
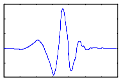
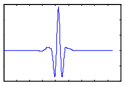
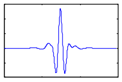


Figure 5-1 Discrete Wavelet packet Analysis [11]

Prior to selecting features for the target classifier it is useful to preprocess the original signal. Any operation that increases our ability to separate the classes is desirable. In this chapter feature selection is based on transformations derived from wavelets in Table 5-1. Training and test sets were constructed using each of the functions. The utility of each of these wavelets for enhancing performance of a classifier was then analyzed. An example of the power of a wavelet transformation is illustrated in Figure 5-2 using the Haar wavelet transform on the original signal.

Table 5-1 Wavelet Functions Used in Wavelet Transform

Db1 Haar Bior1.1		Bior3.5		Db5		Coif4	
Bior1.3		Bior3.7		Db6		Coif5	
Bior1.5		Bior3.9		Db7		Sym2	
Bior2.2		Bior4.4		Db8		Sym3	
Bior2.4		Bior5.5		Db9		Sym4	
Bior2.6		Bior6.8		Db10		Sym5	
Bior2.8		Db2		Coif1		Sym6	
Bior3.1		Db3		Coif2		Sym7	
Bior3.3		Db4		Coif3		Sym8	

In Figure 5-2 the original signal is contained in the first 128 feature index points. The coefficients of the Haar transform are contained in the remaining feature index points. A perfect feature would have a cluster size of 60. This means that this one feature could correctly classify all 60 training signals of that one target class. The original signal features show that the largest number of signals in the training set that can be classified by a single feature is 20 out of a maximum of 60. Selecting a single feature from the wavelet coefficients, it is possible to classify 50 out of 60 signals. This is a significant improvement!

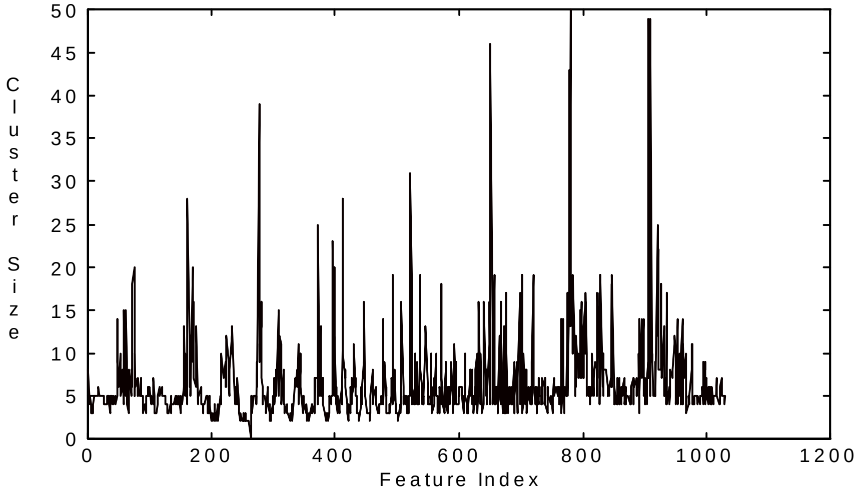


Figure 5-2 Maximum Cluster Sizes

5.3 Wavelet Family Dependence

As observed in the prior discussion, a wavelet transform improves feature selection for target recognition. The natural question is to identify which wavelet family improves target recognition the most. In this section it is demonstrated that there is no single wavelet family that out performs all others in this task.

Proposition 1: *No single wavelet family transform has a statistically significant advantage over any other wavelet family in extracting features for target classification.*

To verify Proposition 1 classifiers were constructed using training sets from all the wavelet families. Table 5-2 shows the results obtained testing the classifier built from the original signal and the associated wavelet transform. In addition, the mean and standard deviation of P_{cc} for the wavelet family are presented in Table 5-2. To determine if there is any significant difference between the families hypothesis testing of the means [19] is used. The mean, μ, and the standard deviation, σ, of the population are calculated using:

$$\mu = \frac{1}{n_w} \sum_{i=1}^{n_w} P_{cc_i} \qquad (5\text{-}2)$$

$$\sigma = \sqrt{\frac{n_w \sum_{i=1}^{n_w} P_{cc_i}^2 + \left(\sum_{i=1}^{n_w} P_{cc_i} \right)^2}{n_w n_w - 1}} \tag{5-3}$$

where μ is the mean value and σ is the standard deviation of probability of correct classification for various wavelet transforms from a selected wavelet family and n_W is the number of wavelets in the selected family. When the mean and standard deviation are computed from samples, μ is replaced by $\bar{x}$ and σ is replaced by s respectively. The hypothesis;

$$\textbf{H0}: \mu_1 = \mu_2 \tag{5-4}$$

is being tested against the alternative hypothesis;

$$\textbf{H1}: \mu_1 \neq \mu_2 . \tag{5-5}$$

The test statistic is computed as follows:

$$Z = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{5-6}$$

H0 is rejected if $|Z| > 1.96$ (1.96 is for a two-tailed test where the results are significant at a level of 0.05). The results of this hypothesis testing are presented in Table 5-3. Table 5-3 shows the test results of the null hypothesis, H0, for means of various wavelet families based on the sample means and standard deviations from Table 5-2.

From the analysis presented, the null hypothesis, that there is no difference in the mean values, must be accepted. This means that there is no statistically significant difference in the performance of the classifiers when different families of wavelets are used to transform the input data. Since there is no difference between the families the question arises is there significant difference within each family? Since the standard deviations are small, 1.4% to 3.8%, it appears that there is no significant difference between the wavelets within the families. It is safe to conclude that classifier performance would be the same no matter which wavelet was chosen. It would be most efficient (from a computational standpoint) to use the simplest wavelet possible. Therefore it is best to use the Db1 (Haar) wavelet.

Table 5-2 Performance of Wavelets

Name	P_{cc}	Name	P_{cc}	Name	P_{cc}	Name	P_{cc}
Bior1.3	0.72488	Db1 (Haar)	0.77130	Coif1	0.76115	Sym2	0.79153
Bior1.5	0.78045	Db2	0.79576	Coif2	0.78231	Sym3	0.75886
Bior2.2	0.75760	Db3	0.75886	Coif3	0.77133	Sym4	0.77458
Bior2.4	0.78150	Db4	0.79160	Coif4	0.78770	Sym5	0.75800
Bior2.6	0.77400	Db5	0.77567	Coif5	0.76943	Sym6	0.76345
Bior2.8	0.78600	Db6	0.78120			Sym7	0.76591
Bior3.1	0.70550	Db7	0.77460			Sym8	0.78275
Bior3.3	0.77030	Db8	0.76760				
Bior3.5	0.78020	Db9	0.79410				
Bior3.7	0.79410	Db10	0.77630				
Bior3.9	0.79290	Db11	0.75598				
Bior4.4	0.72990	Db12	0.76300				
Bior5.5	0.74150						
Bior6.8	0.73010						
Mean	0.76064		0.77550		0.77438		0.77073
Std. Dev.	0.02890		0.01329		0.01060		0.01272

Table 5-3 Wavelet Family Hypothesis Test

Wavelet Name	Wavelet Name	$\lvert Z \rvert$	Accept or Reject H0
Biorthogonal	Daubechies	1.723060	Accept
Biorthogonal	Coiflet	1.516130	Accept
Biorthogonal	Symlet	1.109050	Accept
Daubechies	Coiflet	0.183654	Accept
Daubechies	Symlet	0.775505	Accept
Coiflet	Symlet	0.540601	Accept

Normally this type of analysis is limited to large samples where the standard deviations of the samples are known. This is not the case in this analysis so additional hypothesis testing must be performed.

A t-test is used when either or both of the populations are small and the population variances are unknown. However, it must be assumed that the standard deviations of both populations are the same. The t statistic is defined as:

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}} \qquad \textbf{(5-7)}$$

and the null hypothesis, H0, is rejected if the value of t is greater than 2.074 or 2.85 depending on the level of significance chosen. The t-test was performed on different pairs of wavelet families, Table 5-4, and gives the same results as before. This indicates that the small number of samples did not give a false acceptance of H0.

Table 5-4 Wavelet Family Comparison - Hypothesis t-Test

Wavelet Family	Wavelet Family	t	Accept or Reject H0
Biorthogonal	Daubechies	-1.6356	Accept
Biorthogonal	Coiflet	-1.0226	Accept
Biorthogonal	Symlet	-0.87359	Accept
Daubechies	Coiflet	0.166612	Accept
Daubechies	Symlet	0.766102	Accept
Coiflet	Symlet	0.523066	Accept

5.4 Feature Size Dependence

When constructing the classifier, there are times when the classifier is selecting features to classify just a few training signals. However, each new feature increases the dimensionality of the statistical feature space in which the signal classification is performed. The increase in space dimensionality reduces the accuracy of the statistical representation of the training data. As a result, it is possible that when the classifier is choosing a feature to classify a few signals, the classifier performance may decrease on the test set. Based on this the following proposition is made:

Proposition 2: *Features which classify a small number of training signals do not significantly improve classifier performance.*

An analysis was performed to determine what size feature (the number of signals that were classified from the training set) could be safely ignored. Elimination of features classifying a small number of signals allows for much faster training of the classifier and better generalization. Table 5-5 shows the results of the analysis for the Daubechies wavelet family. Each column in the table shows the probability of correct classification with the minimum feature size eliminated as indicated in the first row.

Using the Z test (Equation 5-6), as before, it was found that eliminating features that classify only one signal produces a statistically significant difference in the classifiers. However, it was suspected that because of the small sample size it might be necessary to perform a t-test (Equation 5-7). As seen in Table 5-5, all features which classify less than five training signals can be safely eliminated.

The t-test depends on the standard deviations being equal so a test for this was also made. This is accomplished by using the statistic

$$F = \frac{s_1^2}{s_2^2} \qquad \textbf{(5-8)}$$

If this value is less than 2.85 the null hypothesis, that the variances are equal, must be accepted. As seen from Table 5-5, the null hypothesis for all the tests (the standard deviations are equal) can be accepted.

The results for the other families of wavelets are presented in Tabl e5-6 through Tabl e5-8. The analysis was performed as before. These results show that no matter what family of wavelet is used there is no statistical performance difference between the classifiers when features that classify four signals or less are removed from the classifier. Removing these features significantly reduces the time required to create the classifier.

Table 5-5 Significance of Eliminating Features of Daubechies Wavelets

Name	Test P_{cc}	>1 P_{cc}	>2 P_{cc}	>3 P_{cc}	>4 P_{cc}	>5 P_{cc}	>6 P_{cc}	>7 P_{cc}
Db1	0.77130	0.76903	0.75369	0.71595	0.67470	0.64463	0.61022	0.58327
Db2	0.79576	0.79474	0.77753	0.75514	0.71347	0.67884	0.65625	0.62203
Db3	0.75886	0.75762	0.73772	0.70496	0.67241	0.64131	0.61124	0.58242
Db4	0.79160	0.79060	0.77318	0.73483	0.70124	0.67408	0.65355	0.62640
Db5	0.77567	0.77443	0.75888	0.72965	0.69026	0.65667	0.63760	0.61002
Db6	0.78120	0.78065	0.76759	0.73462	0.69689	0.67035	0.62702	0.60214
Db7	0.77460	0.77381	0.75784	0.72695	0.69378	0.66351	0.63739	0.61624
Db8	0.76760	0.76655	0.74748	0.71327	0.68113	0.63822	0.61313	0.58058
Db9	0.79410	0.79247	0.77712	0.74913	0.70974	0.68175	0.65666	0.62411
Db10	0.77630	0.77526	0.75764	0.72073	0.68549	0.65522	0.62993	0.61210
Db11	0.75598	0.75452	0.73192	0.69771	0.65687	0.62743	0.58783	0.56087
Db12	0.76300	0.76157	0.74456	0.71409	0.66391	0.64318	0.61478	0.56875
Mean	0.77550	0.77427	0.75710	0.72475	0.68666	0.65627	0.62797	0.59908
Std. Dev.	0.01329	0.01340	0.01496	0.01708	0.01762	0.01762	0.02146	0.02285
Z		0.22515	3.18521	8.12297	13.94261	18.71136	20.24809	23.11965
Alpha=.05	1.96	NOT sig.	Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Alpha=.01	2.58	NOT sig.	Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
t s1=s2,normal		0.02601	0.37924	1.00870	1.75033	2.34906	2.74159	3.21468
t .05/22	2.074	NOT sig.	NOT sig.	NOT sig.	NOT sig.	Sig.	Sig.	Sig.
t .01/22	2.819	NOT sig.	NOT sig.	NOT sig.	NOT sig.	NOT sig.	NOT sig.	Sig.
F		1.00785	1.12539	1.28459	1.32554	1.32564	1.61402	1.71875
Alpha=.05 2.85 (11,12)		Accept	Accept	Accept	Accept	Accept	Accept	Accept
Alpha=.01 4.54 (11,10)		Accept	Accept	Accept	Accept	Accept	Accept	Accept

Table 5-6 Significance of Eliminating Features for Symlet Wavelets

Name	Test	>1	>2	>3	>4	>5	>6	>7
	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}
Sym2	0.79153	0.79039	0.77297	0.74601	0.70704	0.68008	0.64691	0.61726
Sym3	0.75886	0.75762	0.73772	0.70496	0.67241	0.64131	0.61124	0.58242
Sym4	0.77458	0.77277	0.75763	0.72985	0.69129	0.65314	0.60380	0.58390
Sym5	0.75800	0.75701	0.73855	0.71056	0.67428	0.64194	0.62079	0.58409
Sym6	0.76345	0.76261	0.74665	0.71949	0.69461	0.65853	0.61811	0.59883
Sym7	0.76591	0.76509	0.74830	0.71927	0.67822	0.63571	0.60585	0.58118
Sym8	0.78275	0.78148	0.76801	0.73919	0.70975	0.67762	0.63906	0.61418
Mean	0.77073	0.76957	0.75283	0.72419	0.68966	0.65548	0.62082	0.59455
Std. Dev	0.01272	0.01261	0.01384	0.01492	0.01526	0.01773	0.01646	0.01564
Z		0.22402	3.29655	8.22262	14.13325	18.29834	24.95777	30.26778
Alpha=.05	1.96	NOT Sig.	Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Alpha=.01	2.58	NOT Sig.	Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
t s1=s2,normal		0.02521	0.38028	0.96966	1.67871	2.28800	3.03959	3.62363
t .05/22	2.074	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	Sig.	Sig.	Sig.
t .01/22	2.819	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	Sig.	Sig.
F		1.00860	1.08822	1.17248	1.19974	1.39325	1.29408	1.22955
Alpha=.05	2.85	Accept	Accept	Accept	Accept	Accept	Accept	Accept
Alpha=.01	4.54	Accept	Accept	Accept	Accept	Accept	Accept	Accept

Table 5-7 Significance of Eliminating Features for Coiflet Wavelets

Name	Test	>1	>2	>3	>4	>5	>6	>7
	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}	P_{cc}
Coif1	0.76115	0.75991	0.74104	0.71139	0.67635	0.64028	0.60566	0.57041
Coif2	0.78231	0.78147	0.76551	0.72819	0.69896	0.67346	0.63241	0.60877
Coif3	0.77133	0.77008	0.75225	0.72675	0.69461	0.65937	0.62371	0.59240
Coif4	0.78770	0.78625	0.77008	0.73835	0.70456	0.68030	0.66371	0.62743
Coif5	0.76943	0.76862	0.74954	0.71699	0.68859	0.65541	0.63386	0.60753
Mean	0.77438	0.77327	0.75568	0.72433	0.69261	0.66176	0.63187	0.60131
Std. Dev	0.01060	0.01056	0.01191	0.01047	0.01081	0.01572	0.02105	0.02128
Z		0.25889	4.06272	11.63783	18.70997	20.58158	20.95222	25.22373
Alpha=.05	1.96	NOT Sig.	Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Alpha=.01	2.58	NOT Sig.	Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
t s1=s2,normal		0.02663	0.43176	1.19448	1.93590	2.40503	2.77536	3.35826
t .05/22	2.074	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	Sig.	Sig.	Sig.
t .01/22	2.819	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	Sig.
F		1.00351	1.12434	1.01191	1.02042	1.48319	1.98604	2.00790
Alpha=.05	2.85	Accept	Accept	Accept	Accept	Accept	Accept	Accept
Alpha=.01	4.54	Accept	Accept	Accept	Accept	Accept	Accept	Accept

Table 5-8 Significance of Eliminating Features for Biorthogonal Wavelets

Name	Test P_{cc}	>1 P_{cc}	>2 P_{cc}	>3 P_{cc}	>4 P_{cc}	>5 P_{cc}	>6 P_{cc}	>7 P_{cc}
Bior1.3	0.72488	0.72466	0.71160	0.68548	0.64961	0.61934	0.58160	0.55175
Bior1.5	0.78045	0.77899	0.76178	0.72985	0.68486	0.65355	0.62349	0.60048
Bior2.2	0.75760	0.75639	0.73255	0.70332	0.65377	0.62744	0.59385	0.56337
Bior2.4	0.78150	0.77961	0.76365	0.73047	0.69253	0.66288	0.63697	0.59634
Bior2.6	0.77400	0.77194	0.75577	0.72591	0.69584	0.65874	0.64091	0.60566
Bior2.8	0.78600	0.78397	0.76551	0.73006	0.69482	0.66248	0.63345	0.59551
Bior3.1	0.70550	0.70455	0.68008	0.64484	0.60440	0.56750	0.53391	0.49556
Bior3.3	0.77030	0.76944	0.75182	0.71823	0.68195	0.64381	0.61188	0.58534
Bior3.5	0.78020	0.77899	0.76406	0.73462	0.69461	0.66807	0.63883	0.62515
Bior3.7	0.79410	0.79226	0.77567	0.75224	0.71783	0.69316	0.65936	0.62120
Bior3.9	0.79290	0.79247	0.77837	0.74789	0.70891	0.68361	0.65189	0.62349
Bior4.4	0.72990	0.72944	0.71783	0.69067	0.66848	0.62744	0.60753	0.58473
Bior5.5	0.74150	0.74146	0.73192	0.70869	0.68755	0.65167	0.61622	0.59736
Bior6.8	0.73010	0.73006	0.72032	0.69565	0.66538	0.63553	0.61936	0.58680
Mean	0.76064	0.75959	0.74364	0.71414	0.67861	0.64680	0.61780	0.58805
Std Dev	0.02890	0.02844	0.02857	0.02855	0.02887	0.03113	0.03232	0.03386
Z		0.08970	1.44924	3.96527	6.95551	9.28304	11.41263	13.42897
Alpha=.05	1.96	NOT Sig.	NOT Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
Alpha=.01	2.58	NOT Sig.	NOT Sig.	Sig.	Sig.	Sig.	Sig.	Sig.
t s1=s2,normal		0.01518	0.24566	0.67206	1.18217	1.60943	1.99979	2.38636
t .05/22	2.074	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	Sig.
t .01/22	2.819	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.	NOT Sig.
F		1.01615	1.01172	1.01229	1.00088	1.07728	1.11825	1.171765
Alpha=.05 2.85 (11,12)		Accept	Accept	Accept	Accept	Accept	Accept	Accept
Alpha=.01 4.54 (11,10)		Accept	Accept	Accept	Accept	Accept	Accept	Accept

5.5 Iterated Wavelet Transform

A single wavelet transform using any of the previous families improves classification performance. Is it possible to improve upon this performance? If the original signals are transformed and then 128 of the most informative pseudo range bins selected for further transformation, a new linear transformation of the input data is created [20]. Therefore, it is proposed:

Proposition 3: *By iteratively selecting the most informative pseudo range bins and transforming them the informative value of the range bins increase and therefore yields a better classifier.*

An experiment was performed to verify this proposition. The original 128 range bin signal was transformed (using the Haar wavelet) as previously discussed creating 1024 pseudo range bins. A box classifier was constructed. The range bins used as features were chosen for further transformation. If there are more than 128 pseudo range bin features, then the features that classify the most training signals are selected. If there are fewer than 128 features, then additional pseudo range bins are selected from the middle of the pseudo signal. These 128 range bins are wavelet transformed and a classifier is constructed and tested. This procedure is repeated a total of 12 times and the results are presented in Table 5-9 and Figure 5-3.

When using just one wavelet transform on the original signal, Stirman showed an increase in P_{cc} of 6 percentage points [6] and 7.53 percentage points over the baseline classifier [7]. This difference, over the baseline classifier, may have resulted from changing the type of classifier or the use of wavelets. Stirman did not attribute the increase in performance to one or the other, neither did he analyze the significance of using the wavelet transform. In the results presented here, it was found that using the same classifier, the improvement in P_{cc} after one application of the wavelet transform is 4.2 percentage points. This improvement is smaller than the one observed by Stirman but this difference can be attributed to the use of a different classifier. The most important curve in Figure 5-3 is the one for P_{cc}. This curve shows an increase in overall classifier performance from 0.7713 to 0.89717 by iteration 10. This represents an improvement of 12 percentage points. Furthermore, Target 2 improved by 25 percentage points and Target 6 by 18 percentage points. This is a significant improvement in performance over a single wavelet transform and confirms the benefit of using the iterated wavelet transform.

Table 5-9 Results of Iterative Application of Haar Transform

Iteration	P_{cc}	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
0	.7713	.9490	.6219	.8219	.6853	.8134	.7363
1	.81361	.9552	.6741	.8555	.6692	.8893	.8383
2	.84762	.9552	.7724	.9128	.7027	.9104	.8321
3	.86421	.9453	.7823	.9452	.7363	.9154	.8607
4	.87976	.9453	.7935	.9465	.7774	.9328	.8831
5	.88618	.9453	.8172	.9552	.7550	.9316	.9129
6	.88453	.9453	.8197	.9601	.7376	.9316	.9129
7	.89095	.9391	.8507	.9664	.7450	.9316	.9129
8	.89261	.9391	.8246	.9664	.7799	.9316	.9142
9	.88867	.9391	.8346	.9651	.7488	.9316	.9129
10	.89717	.9391	.8706	.9639	.7512	.9391	.9192
11	.89365	.9353	.8570	.9639	.7512	.9391	.9154
12	.90049	.9353	.8483	.9689	.7749	.9465	.9291

A question arose as to why there would be a decrease in performance on some of the targets such as seen on Target 2 between iterations 7 and 8. It is apparent that the iterated wavelet transforms yield an increasing performance in the entire classifier. Individual targets may sacrifice performance while overall performance increases. In general, the momentary decreases are recovered in later iterations. This may be a manifestation of the biasing problem reported by Stirman [7]. If so, by iterating the wavelet transform this problem appears to either be mitigated or eliminated. For this problem the maximum advantage of iterating the wavelets happens at about 10 iterations.

Performance increase was anticipated as fewer features are required and the features that are chosen classify more signals. This supposition was confirmed by experiment and is graphically illustrated in Table 5-4. It is easily seen that the targets with the fewest features have the highest performance (Table 5-3). The targets with the lowest performance require the most features. This happens when two targets are very similar and the same range bins contain the information required to segment the targets. In such cases an individual feature provides little differentiation between targets and additional features are required. It should be noted that performance on a given target improves most significantly when the number of features required for classification decreases as in Target 2, Target 5, and Target 6.

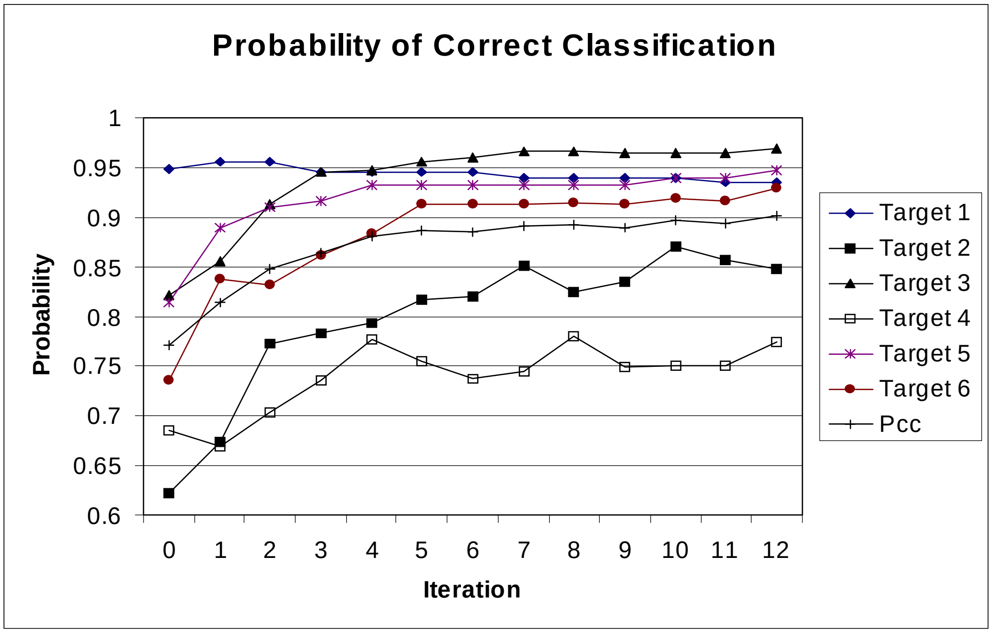


Figure 5-3 Iterated Wavelet Results

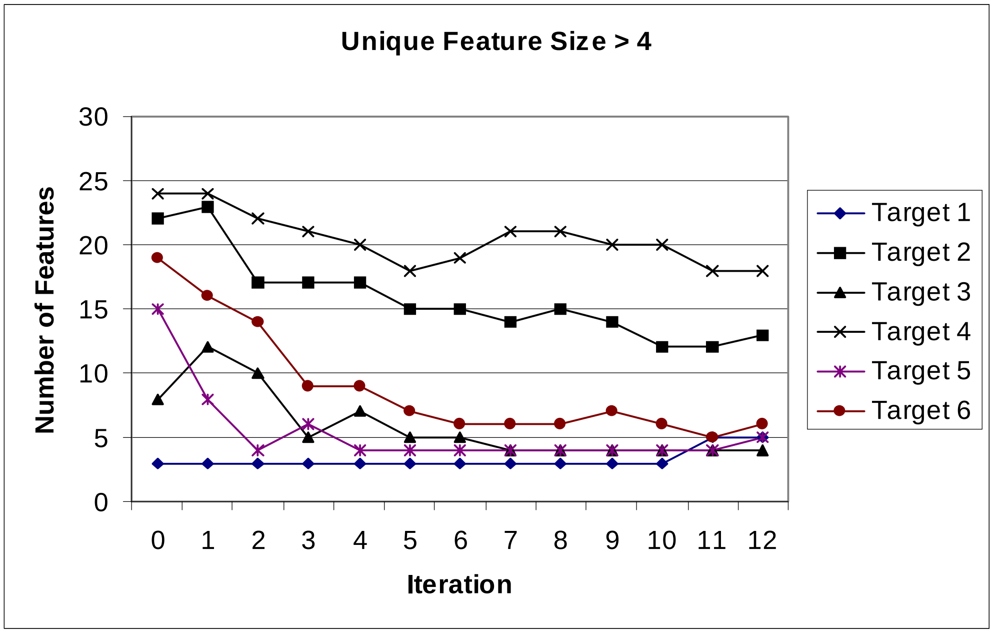


Figure 5-4 Feature Sizes

5.6 Summary

In this chapter it was shown that for the HRR target recognition problem the use of wavelets to enrich the feature space improves classifier performance. In addition, it was shown that there is no statistically significant difference in performance of the classifier when different wavelets are chosen. This means that the simplest wavelet implementation will do as good a job as any other wavelet, at least for the HRR target recognition problem. Results were presented that show that (using a box classifier) features that classify fewer than five training signals can safely be ignored without producing a statistically significant change in the classifier's performance.

The most significant contribution of this chapter is the idea of using iterated wavelet transforms to improve classifier performance. Information content methods to down select the pseudo range bins, instead of using the box classifier features, were presented in Chapter 4. This information-based method has shown similar improvement in classification performance.

The application of the iterative wavelet method used here to improve performance could potentially be used in other problem domains.

6 Rough Set Classification

As previously discussed in Chapter 2, rough set theory (RST) is not capable of being used on very large problems such as the HRR classification problem used in this research. The initial problem is large considering the number of signals in the training set and 128 range bins. By expanding the number of range bins through wavelet feature generation to a total of 1,024 the problem becomes intractable for RST. Therefore, in this chapter methods are developed for selecting an appropriate subset of range bins for RST analysis and classifier development. Since this is only a subset of the full information system RST must be extended to include this kind of partitioning.

Even the partitioned data is much too large for conventional RST analysis. In this Chapter a new algorithm for computing reducts is introduced. This algorithm is an O(n^2) algorithm which is vastly superior to the non-polynomial methods currently available in RST.

6.1 Definition of Focused Reducts

The data set partitioning scheme was discussed in Section 4.2. The reasons and advantages of this partitioning were discussed in that section also. This partitioning results in the original information system (U, A) being transformed to a **focused information system** (U, B) that represents local properties of the information system. A **focused reduct** F is a reduct of the focused information system where Ind(F) = Ind(B). A focused reduct in general is not a reduct of the original system as it may not differentiate all objects and in general $\mathrm{Ind}(A) \subseteq \mathrm{Ind}(B)$.

6.2 Definition of Power Information System

The **power information system** is defined as a set of all focused information systems.

$$P(U, A) = \{(U, B) : B \in 2^A\} \quad \textbf{(6-1)}$$

In other words the power information system of a given information system (U, A) is a super set of information systems defined on the power set of A. The power information system is more robust than the original information system and can extract useful knowledge from incomplete or corrupted data.

6.3 Definition of Covered Information system

Using the previous definitions it is now possible to define **covered information system** as:

$$C(U, A) = \left\{(U, B) : B \in C \subseteq 2^A \wedge A \subseteq \bigcup_{B \in C} B\right\} \quad \textbf{(6-2)}$$

In order to reduce computational cost focused reducts will be chosen from a covered information system. In general a covered information system is redundant which means that $\mathrm{card}(A) < \sum_i \mathrm{card}(A_i)$. This redundancy is what creates the more robust classifier. However, a means must be developed to properly amalgamate or fuse this data into a classifier. This will be discussed in Section 6.6.

Conjecture. A covered information system yields a combined classification performance of focused reducts exceeding performance of the reducts of the original information system. In addition, the obtained classification is more robust to signal distortion and can work with partially determined signals.

The correctness of this conjecture is addressed in Section 7.1

6.4 Determination of Reducts

The method of determining reducts as described in Chapter 2 are too computationally intensive to support a problem of the size used in this dissertation. In this section a new methodology is presented with significantly enhanced performance. Once the data has been partitioned and labeled the following procedure is used. The first step is reducing the number of range bins considered. Next a consistent training set must be created. Once this is accomplished it is possible to calculate the core and reducts.

6.4.1 Selection of Salient Range Bins

Even with efficient methods of determining near minimal reducts the enhanced feature set generated in this research is still too large. It was determined that by using uncompiled MATLAB computer code running on a 400 MHz processor that a reasonable number of attributes to be considered would be approximately 50.

In Section 4.5 an entropy index was developed which provides information as to the value of a given range bin in the process of separating the various target classes. The higher the index value the more valuable the range bin. The 50 range bins with the highest entropy index will be selected for the determination of the reducts.

Figure 6-1 through Figure 6-29 are the graphs of the entropy index for each partition. The data is sorted in descending entropy order so that the effect of selecting the 50 highest entropy index range bins can be determined. Obviously the partitions with more range bins show a greater spread in the entropy index, as in Figure 6-1, where all the range bins are in the partition. It is interesting to note the partitions which have little information, Figure 6-6, Figure 6-9, Figure 6-14, Figure 6-15, Figure 6-20, and Figure 6-21. These partitions show little value in the classification process which will be shown later in Table 7-2. In these cases the 50 highest entropy index range bins contain most of the information.

It is important to note that the partitions which are based on the interleave partitioning tend to spread the information as expected across the range bins. It should also be noted that in every case the graphs tend to be generally concave or at the very least a steep negative slope meaning that as more range bins are selected the benefit is decreasing at an ever higher rate. This effect is most pronounced in figures like Figure 6-3 and Figure 6-8. In partitions of small size selecting the first 50 bins results in having virtually all the information in the partition. This would suggest that a scheme where the number of range bins selected for reduct generation would be a function of the total range bins in the partition and the area under the entropy index curve should be explored.

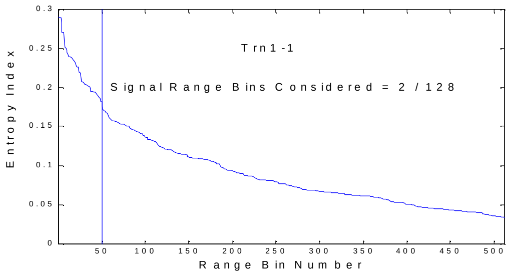


Figure 6-1 Partition Trn1-1 Entropy

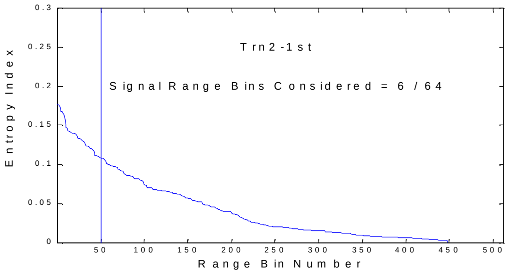


Figure 6-2 Partition Trn2-1st Entropy

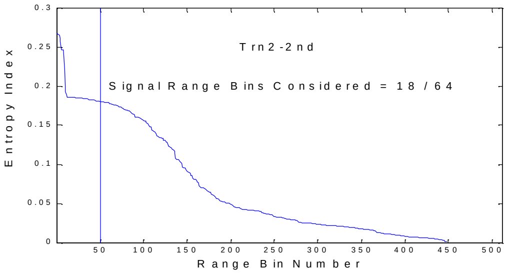
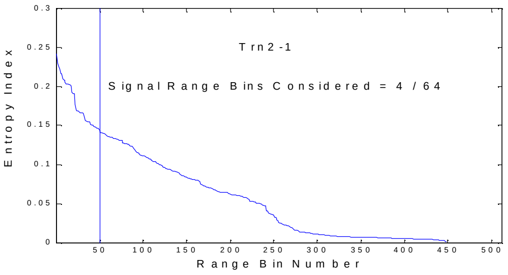


Figure 6-3 Partition Trn2-2nd Entropy



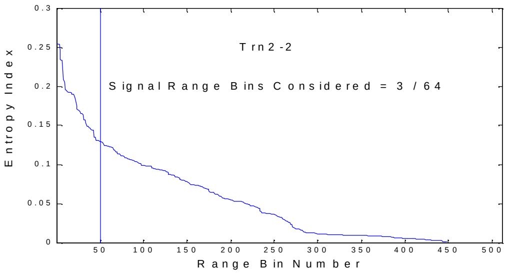
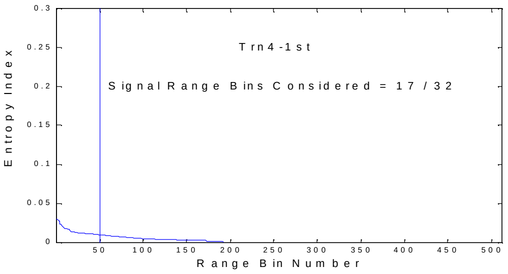


Figure 6-4 Partition Trn2-1 Entropy

Figure 6-5 Partition Trn2-2 Entropy



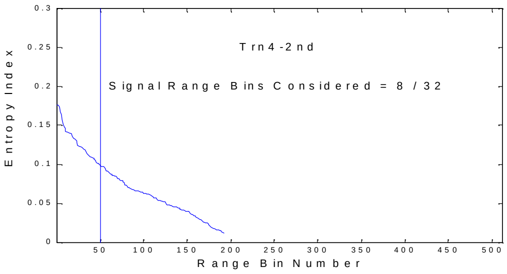
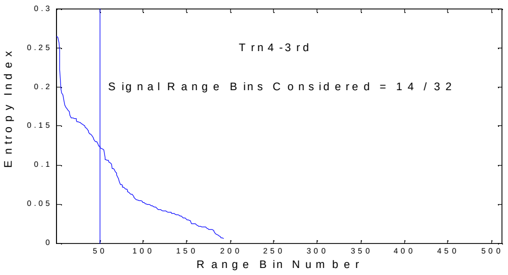


Figure 6-6 Partition Trn4-1st Entropy

Figure 6-7 Partition Trn4-2nd Entropy



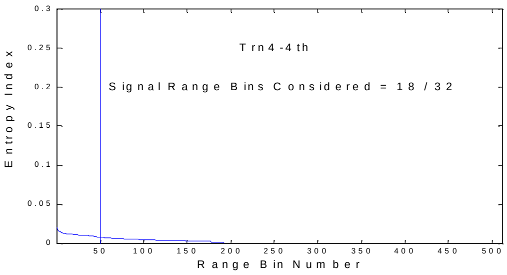
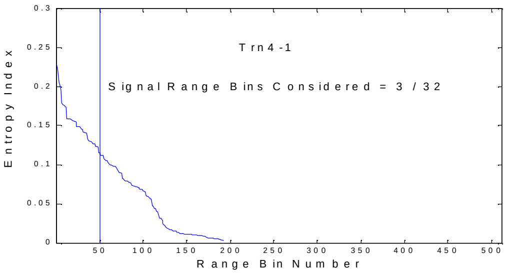


Figure 6-8 Partition Trn4-3rd Entropy

Figure 6-9 Partition Trn4-4th Entropy



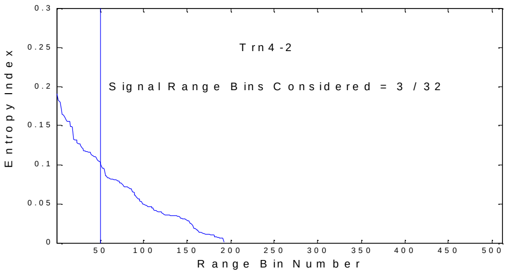


Figure 6-10 Partition Trn4-1 Entropy

Figure 6-11 Partition Trn4-2 Entropy

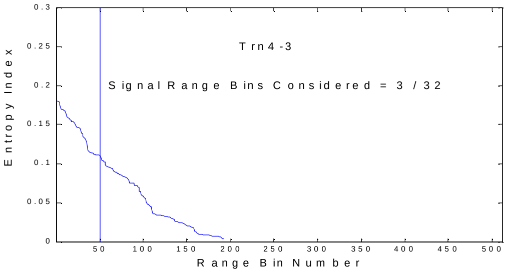


Figure 6-12 Partition Trn4-3 Entropy

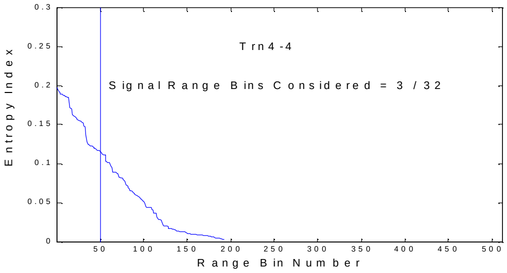


Figure 6-13 Partition Trn4-4 Entropy

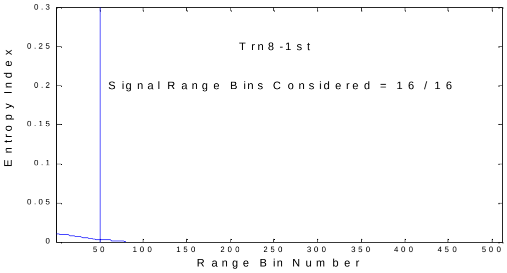


Figure 6-14 Partition Trn8-1st Entropy

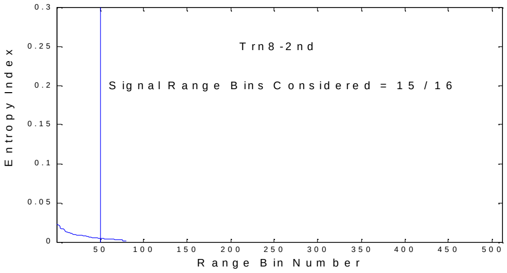


Figure 6-15 Partition Trn8-2nd Entropy

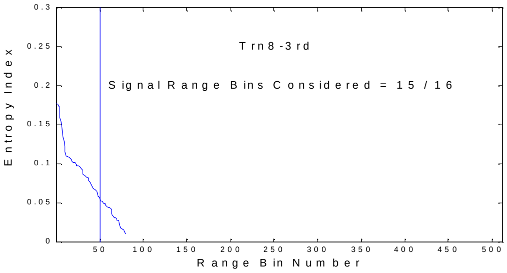


Figure 6-16 Partition Trn8-3rd Entropy

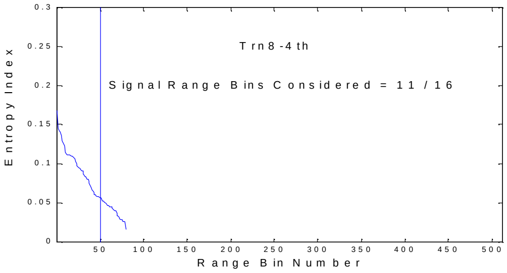


Figure 6-17 Partition Trn8-4th Entropy

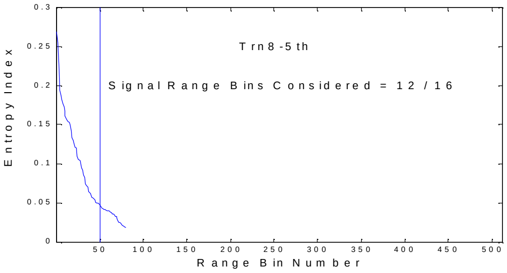


Figure 6-18 Partition Trn8-5th Entropy

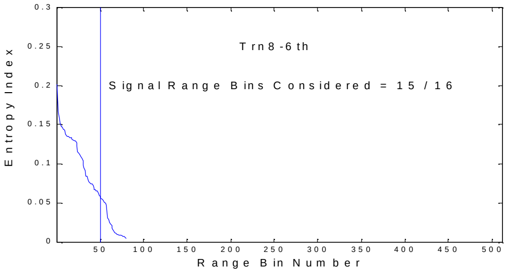


Figure 6-19 Partition Trn8-6th Entropy

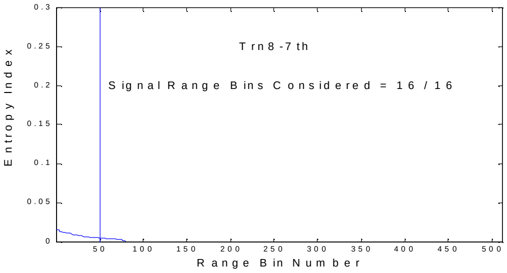


Figure 6-20 Partition Trn8-7th Entropy

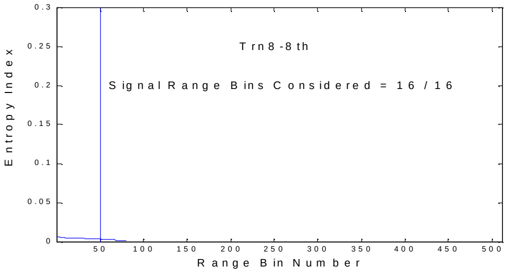


Figure 6-21 Partition 8-8th Entropy

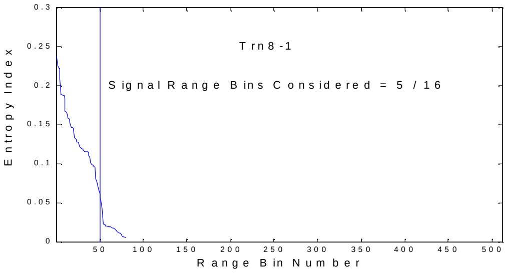


Figure 6-22 Partition Trn8-1 Entropy

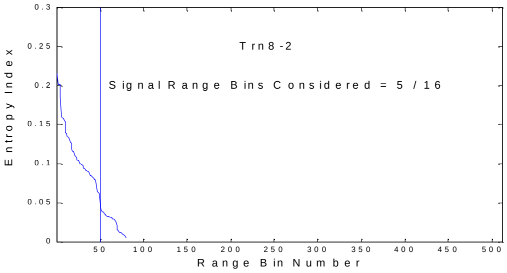


Figure 6-23 Partition Trn8-2 Entropy

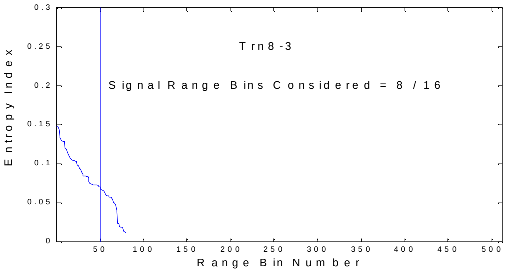


Figure 6-24 Partition Trn8-3 Entropy

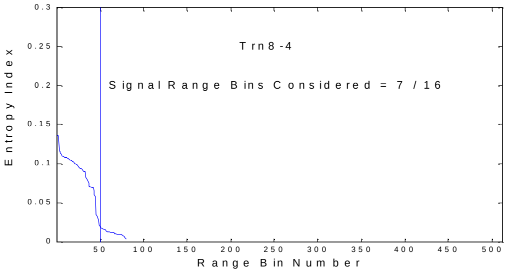


Figure 6-25 Partition Trn8-4 Entropy

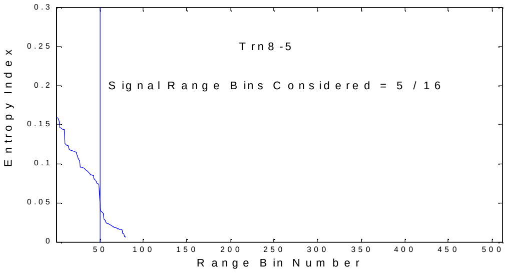


Figure 6-26 Partition Trn8-5 Entropy

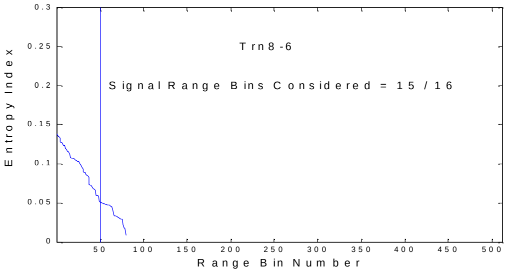


Figure 6-27 Partition Trn8-6 Entropy

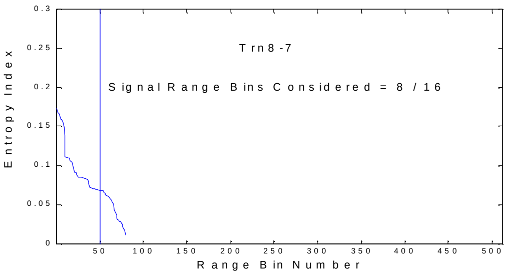


Figure 6-28 Partition Trn8-7 Entropy

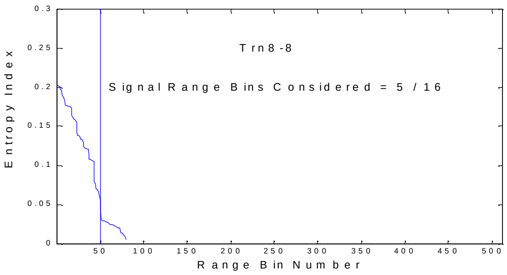


Figure 6-29 Partition Trn8-8 Entropy

6.4.2 Elimination of Duplicates

The first step toward efficiency is to eliminate duplicates from the training set. A **duplicate** is defined as:

$$x_i = x_j \text{ and } c_i = c_j \qquad \textbf{(6-3)}$$

which simply means that two signals of the same class have the same valued attributes. Basically duplicates add no additional information while increasing computational costs and all but one can be eliminated. Table 6-1 shows the number of duplicates for each partition and each target class in the training set.

Table 6-1 Number of Duplicate Signals

Partition Name	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Total # of Duplicates
Trn1-1	226	189	103	126	161	104	909
Trn2-1	214	146	112	135	100	59	766
Trn2-2	178	144	84	43	138	92	679
Trn2-1st	117	41	6	5	33	33	235
Trn2-2nd	225	226	248	213	200	201	1313
Trn4-1	161	142	53	92	81	74	603
Trn4-2	146	95	69	43	93	110	556
Trn4-3	166	100	140	95	122	78	701
Trn4-4	185	158	152	83	122	132	832
Trn4-1st	157	179	131	118	98	94	777
Trn4-2nd	65	17	0	0	16	9	107
Trn4-3rd	88	150	141	74	86	125	664
Trn4-4th	192	162	132	170	153	140	949

Table 6-1 Number of Duplicate Signals (Continued)

Partition Name	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Total # of Duplicates
Trn8-1	189	175	93	82	97	151	787
Trn8-2	169	158	84	47	128	128	714
Trn8-3	129	91	112	49	118	94	593
Trn8-4	146	34	110	18	59	51	418
Trn8-5	174	51	101	52	83	55	516
Trn8-6	108	43	65	40	29	121	406
Trn8-7	132	79	50	106	99	120	586
Trn8-8	194	155	49	70	153	150	771
Trn8-1st	237	234	228	228	225	226	1378
Trn8-2nd	174	181	132	133	116	122	858
Trn8-3rd	50	40	22	7	30	191	340
Trn8-4th	3	15	2	3	0	3	26
Trn8-5th	0	9	23	2	34	24	92
Trn8-6th	228	41	22	65	57	207	620
Trn8-7th	242	197	183	224	210	210	1266
Trn8-8th	239	233	215	237	237	239	1400

6.4.3 Elimination of Ambiguities

The next step is to eliminate ambiguities from the training set. An **ambiguity** is defined as:

$$x_i = x_j \text{ where } c_i \neq c_j \qquad \textbf{(6-4)}$$

which means that two signals are identical but represent different classes. Therefore both of the offending signals are removed from the training set. Often duplicates and ambiguities are artifacts of the labeling, partitioning, and bin selection processes. Table 6-2 shows the ambiguities.

Table 6-2 Number of Ambiguities

Partition Name	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
Trn1-1	0	19	10	19	8	15
Trn2-1	5	17	13	19	10	19
Trn2-2	4	23	27	22	18	21
Trn2-1st	0	3	0	3	0	0
Trn2-2nd	19	18	13	18	18	18
Trn4-1	11	21	16	22	10	24

Table 6-2 Number of Ambiguities (Continued)

Partition Name	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
Trn4-2	12	21	18	25	20	15
Trn4-3	16	26	12	18	7	27
Trn4-4	8	28	24	27	22	23
Trn4-1st	71	61	82	88	85	87
Trn4-2nd	0	1	0	1	0	0
Trn4-3rd	20	21	22	22	30	31
Trn4-4th	61	81	88	76	78	85
Trn8-1	11	23	30	27	17	24
Trn8-2	25	18	37	26	24	32
Trn8-3	29	39	31	43	25	48
Trn8-4	35	35	20	29	41	32
Trn8-5	25	48	43	66	41	34
Trn8-6	23	40	33	32	42	23
Trn8-7	18	40	38	36	27	31
Trn8-8	19	25	35	35	26	30
Trn8-1st	27	27	29	30	36	35
Trn8-2nd	83	72	102	105	100	94
Trn8-3rd	12	32	11	14	5	22
Trn8-4th	0	4	0	2	20	0
Trn8-5th	2	5	10	6	21	26
Trn8-6th	19	40	28	39	39	24
Trn8-7th	24	38	48	36	34	43
Trn8-8th	28	26	26	25	26	24

Once all the ambiguities and duplicates are removed the final training signals are ready for the reducts to be found. Table 6-3 shows the number of training signals remaining.

Table 6-3 Number of Training Signals

Partition Name	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Total Signals
Trn1-1	41	59	154	122	98	148	622
Trn2-1	48	104	142	113	157	189	753
Trn2-2	85	100	156	202	111	154	808
Trn2-1st	150	223	261	259	234	234	1361
Trn2-2nd	23	23	6	36	49	48	185
Trn4-1	95	104	198	153	176	169	895
Trn4-2	109	151	180	199	154	142	935
Trn4-3	85	141	115	154	138	162	795

Table 6-3 Number of Training Signals (Continued)

Partition Name	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Total Signals
Trn4-4	74	81	91	157	123	112	638
Trn4-1st	39	27	54	61	84	86	351
Trn4-2nd	202	249	267	266	251	258	1493
Trn4-3rd	159	96	104	171	151	111	792
Trn4-4th	14	24	47	21	36	42	184
Trn8-1	67	69	144	158	153	92	683
Trn8-2	73	91	146	194	115	107	726
Trn8-3	109	137	124	175	124	125	992
Trn8-4	86	198	137	220	167	184	829
Trn8-5	68	168	123	149	143	178	829
Trn8-6	136	184	169	195	196	123	1003
Trn8-7	117	148	179	125	141	116	826
Trn8-8	54	87	183	162	88	87	661
Trn8-1st	3	6	10	9	6	6	40
Trn8-2nd	10	14	33	29	51	51	188
Trn8-3rd	205	195	234	246	232	54	1166
Trn8-4th	264	248	265	262	265	264	1568
Trn8-5th	265	253	234	259	212	217	1440
Trn8-6th	20	186	217	163	171	36	793
Trn8-7th	1	32	36	7	23	14	113
Trn8-8th	0	8	26	5	4	4	47

6.4.4 Calculation of the Core

After all the duplicates and ambiguities are eliminated the next step is to calculate the core. Recall from Chapter 2 that the core is the intersection of all the reducts. Remember that when a discernibility matrix is produced the singleton entries are attributes that form the core. This means that the core can be determined by removing one attribute at a time and then seeing if there are any ambiguities introduced into the training set. If ambiguities are found, that means that the removed attribute is the only attribute able to discriminate between the two signals of different classes. Therefore, it must be in every reduct and therefore part of the core. This operation is of O(n) time complexity.

6.4.5 Calculation of the Reducts

Calculation of the reducts is more complicated and time consuming than calculation of the core. Recall that reducts are required to preserve all the information in the training data. That is, all signals must be correctly classified using only the attributes in the reduct. Let $\hat{B}$ represent the set of all attributes in a partition. Let R represent the set of all reducts for a given partition. Let $r \in R$ represent one of the reducts of the partition and let r represent the attributes currently in the reduct being constructed. Let c represent the core of the partition. Since all reducts must contain the core the process begins from there, $r = c$ and $\hat{B} = \hat{B} - c$. Now for each $b \in \hat{B}$ calculate the number of ambiguities in the training set and call this number b_a. If there are no ambiguities then r is a reduct. If there are ambiguities, then set $r = r \cup \min(b_a)$ and $\hat{B} = \hat{B} - r$. Repeat until there are no ambiguities. If $b_{ai}=b_{aj}$ meaning that the addition of attributes i and j resulted in the same number of ambiguities, add attribute i to r and save attribute j for processing as part of another reduct. When processing saved attributes, the saving process is <u>not</u> performed. Two reducts are equivalent if they contain the same attributes. In other words the ordering of the attributes is not important. In essence what happens if the saving process is performed after the first reduct is determined is that equivalent reducts can be generated. Note that this process must terminate because the use of all the attributes would be a reduct by definition. If the process terminates prior to using all the attributes (i.e., there are no ambiguities), then this is a reduct by definition. The only question remaining is whether this is a minimal reduct. A **minimal reduct** is defined as the reduct(s) of locally minimum cardinality (containing the fewest attributes). By examining Table 6-4 reducts are found as small as two more attributes than the core. Since the core is not a reduct (if it were then there would only be one reduct) the absolute minimum size for a reduct would be one more attribute than the core. However, this was tested and there were no reducts found of this size. Therefore, the minimal reduct could at best be of size two more than the core. This means that the procedure described was able to determine a reduct within two attributes of the smallest possible size. Furthermore when the procedure returns several reducts, typically they are of the same minimal size or close to it. Check Table 6-4 for the largest reduct found in each category. Only in two categories of data partitions, Trn8-1st and Trn8-2nd, are the difference between the smallest and the largest reduct more than one attribute. As shall be shown, these two partitions contain very little information about the problem.

Table 6-4 Reduct Results

Partition Name	Core Size	# of Reducts	Smallest Reduct	Largest Reduct
Trn1-1	**15**	**12**	**25**	**26**
Trn2-1	**20**	**2**	**28**	**28**
Trn2-2	**26**	**5**	**31**	**31**
Trn2-1st	**16**	**11**	**27**	**28**
Trn2-2nd	**14**	**3**	**30**	**30**
Trn4-1	**24**	**31**	**31**	**31**
Trn4-2	**21**	**7**	**31**	**31**
Trn4-3	**18**	**3**	**30**	**31**
Trn4-4	**23**	**13**	**30**	**31**
Trn4-1st	**23**	**13**	**33**	**34**
Trn4-2nd	**3**	**1**	**25**	**25**
Trn4-3rd	**25**	**21**	**32**	**33**
Trn4-4th	**12**	**1**	**23**	**23**
Trn8-1	**24**	**1**	**32**	**32**
Trn8-2	**28**	**9**	**37**	**38**
Trn8-3	**33**	**5**	**39**	**40**
Trn8-4	**29**	**13**	**36**	**37**
Trn8-5	**40**	**1**	**42**	**42**
Trn8-6	**39**	**8**	**42**	**42**
Trn8-7	**37**	**7**	**40**	**40**
Trn8-8	**31**	**8**	**35**	**36**
Trn8-1st	**2**	**14**	**11**	**37**
Trn8-2nd	**5**	**7**	**26**	**32**
Trn8-3rd	**30**	**3**	**32**	**33**
Trn8-4th	**10**	**1**	**28**	**28**
Trn8-5th	**30**	**3**	**35**	**35**
Trn8-6th	**29**	**3**	**34**	**35**
Trn8-7th	**7**	**1**	**17**	**17**
Trn8-8th	**6**	**1**	**12**	**12**

6.5 Method of Classification

Once a reduct has been determined the process of classification can proceed. Given an unknown signal the first step is to normalize and perform the wavelet feature generation resulting in $\bar{x}$. Using the notation $\bar{x}(r)$ to indicate only the attributes of the reduct r from $\bar{x}$

a search is performed to find $\bar{x}(r) = x(r)$ where $x(r)$ are the signals from the consistent training set. Any attributes labeled as “don’t care” from the fuzzification process are not considered in determining equality of the signals. If a match is found, then $c_{x(r)}$ is assigned as the class of the unknown signal. If no match is found, then the target is of the unknown class.

All the reducts of R are tested against the full training set to determine the performance of each reduct. The P_{cc} and P_{dec} for testing against the training data will not necessarily result in a perfect score. This is due to the elimination of ambiguities and duplicates from the data which was used to determine the reducts. The P_{cc} for each reduct is saved for use in the fusion process.

6.6 Merging the Focused Reducts

Once all the reducts have been determined each one is tested against the full training set (all ambiguities and duplicates included) and the performance (P_{cc} and P_{dec}) determined. In some cases a reduct based on a data partition which contains mostly noise will have a low P_{cc} and a high P_{dec}. Even though the performance is low this reduct may be able to be used to improve classifier performance if it is combined in the right way with other reducts.

As described in the previous section, a set of reducts will be generated for each partition. In a sense each of these reducts is a classifier. Some of these classifiers will yield a high probability of correct classification (P_{cc}) while others will yield a low P_{cc}. What is desired is to have a scoring function that will weight the votes of each of the classifiers for each target class. This function provides a score for each target class based upon the performance of each reduct voting for that target class. Some of the properties that are desired of this function are:

If all the P_{cc}(s) are zero, the weight should be zero.

If all the P_{cc}(s) are one, the weight should be one.

If there are several “votes” of low confidence for a given target class, the weight should be higher than any of the low confidence votes.

If there is one high confidence vote and several low confidence votes, the weight should be higher than the highest score.

The total weighting function for a given target class is given as:

$$W_t = 1 - \frac{P_{max} + \left|\sum_{i=1}^{n}(1 - P_i)\right|(1 - P_{max})}{\sum_{i=1}^{n}\frac{1}{1 - P_i + \varepsilon}} \qquad \textbf{(6-5)}$$

Where P_{max} is the maximum P_{cc} of all reducts “voting” for target class i, P_i is the P_{cc} of each “vote” for target class i, n is the number of “votes” for target class i, and ε is a small number to prevent division by zero if all P_i are 1.

In order to determine if the goals for the formula are being met and to get a physical feel for how the weighting formula performs a simple table of examples was created. Table 6-5 illustrates the weight generated for a target class based on the “votes” and their P_{cc}.

Table 6-5 Results of Weighting Formula

P_{cc}(1)	P_{cc}(2)	P_{cc}(3)	P_{cc}(4)	P_{cc}(5)	W_t
0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	1.00	1.00	1.00	1.00
0.10	0.10	0.00	0.00	0.00	0.15
0.80	0.10	0.30	0.00	0.00	0.84
0.80	0.20	0.00	0.00	0.00	0.83

As can be seen from the table, all the desirable features of the function are achieved. The maximum value 1 is achieved if all the values are 1 and the minimum value 0 is achieved if all the values are 0. Further when there are several small values, the W_t value is higher than any of the small values. The W_t is also larger than the largest of the P_{cc} values. When the sum of the P_{cc}(s) are the same, the W_t value is the same.

6.7 Summary

This chapter introduced extensions to the rough set theory by defining new terms and concepts. A means to reduce the number of attributes being considered was presented and illustrated through graphs of the entropy index content of the various partitions.A new method of determining near minimal reducts was presented along with the method of using these reducts to perform classification. Since many reducts were generated and in order to make a robust classifier a fusion formula was developed. The culmination of these technical developments will be tested in the next chapter.

7 Classification Results

The proof of the concept presented for performing classification using rough set theory (RST) comes from the classification results. The process for training the classifier and testing the results is shown in Figure 7-30

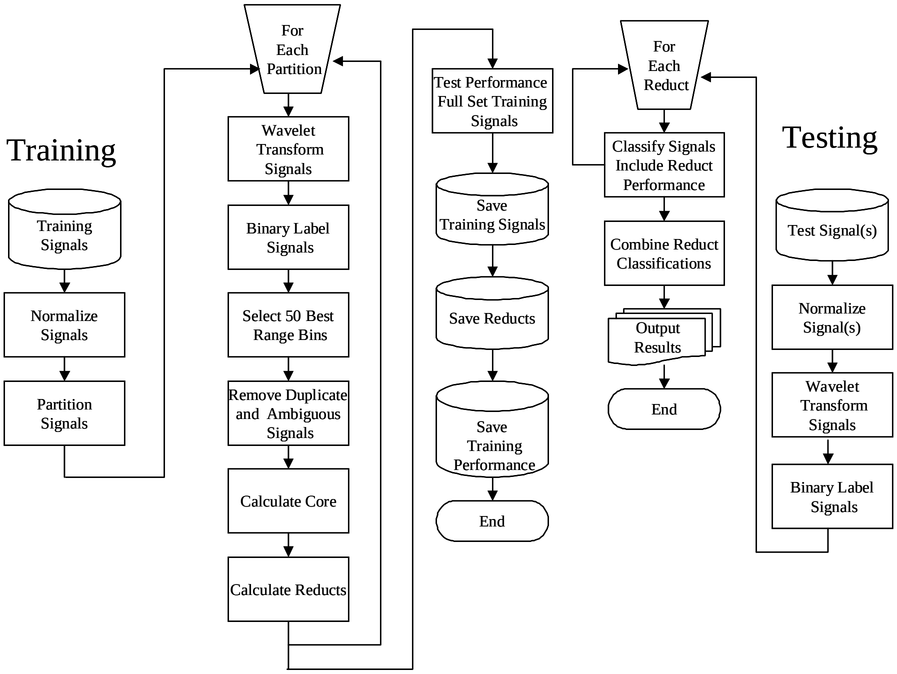


Figure 7-30 Classification and Testing Process

7.1 Basic Focused Reduct Results

The results presented in the following sections are the results of computer simulations. The appendix contains the full output of the rough set classifier which is only summarized in this chapter.

7.1.1 Training Results

The Performance of the classifier on the training data is shown in Table7-1.

Table 7-1 Classifier Performance on Training Data

Partition Name	P_{cc}	P_{dec}	P_{cc}	P_{dec}	P_{cc}	P_{dec}	P_{cc}	P_{dec}
Trn1-1	0.89207	0.93695						
Trn2-1	0.94536	0.83396						
Trn2-2	0.9691	0.82834	0.96991	0.97503				
Trn2-1st	0.99875	0.9975						
Trn2-2nd	0.5505	0.68602	0.9975	0.99938	0.99875	1		
Trn4-1	0.96208	0.83958						
Trn4-2	0.9598	0.88514						
Trn4-3	0.8877	0.87828						
Trn4-4	0.95036	0.69164	0.96998	0.99813				
Trn4-1st	0.79693	0.32584						
Trn4-2nd	0.99938	1						
Trn4-3rd	0.88203	0.78839						
Trn4-4th	0.54545	0.29526	0.99938	1	0.98065	1		
Trn8-1	0.90902	0.75468						
Trn8-2	0.92612	0.70974						
Trn8-3	0.94781	0.66979						
Trn8-4	0.96107	0.7216						
Trn8-5	0.95	0.64919						
Trn8-6	0.95584	0.79151						
Trn8-7	0.90096	0.71848						
Trn8-8	0.97834	0.49001	0.99001	1				
Trn8-1st	0.22995	0.23346						
Trn8-2nd	0.42586	0.36205						
Trn8-3rd	0.98691	0.8583						
Trn8-4th	0.99875	0.99563						
Trn8-5th	0.99012	0.94757						
Trn8-6th	0.7984	0.8608						
Trn8-7th	0.46296	0.23596						
Trn8-8th	0.30547	0.19413	0.99813	1	1	1	1	1

The table has a separate row for each partition. Note that none of the partitions has perfect performance. As discussed earlier, this is due to the elimination of ambiguities and duplicates prior to constructing reducts. For testing purposes all signals are included for performance determination. Going across the columns of the table the first set of P_{cc} and P_{dec} are for the individual partitions. The next set of P_{cc} and P_{dec} use the fusion formula to combine all the reducts of all the partitions of a given type (interleave or block) and size. The next set of columns combines all the reducts of all the classifiers of a given partition size. Finally the last set of columns has the results of combining all the reducts of all the partitions. Note that this final combination results in perfect performance. Generally speaking, ATR systems should have perfect performance on the training set.

7.1.2 Test Results

The results of testing the classifier are shown in Table 7-2. This table is constructed the same way as the table for results on the training data. It is interesting to note that the best classifier performance on any partition group is on the partition with four divisions and the second interleave. In operational scenarios a high P_{cc} of above 90% and a P_{dec} of over 85% is normally required. There is no partition, without using fusion, that even comes close to meeting this criteria. Following the fusion of all partitions the classification performance improves markedly to an acceptable level. Of special interest is that the classifier declares almost 100% of the time. This is very unusual in that the classifier has very good performance even at this level of declaration.

There are some partitions which have a P_{cc} of 0. As expected, this occurs with the smaller partitions and is located at the beginning and ends of the signals which contain mostly noise.

What these results reveal is that it is possible to construct a classifier using rough set theory. This classifier is able to achieve acceptable performance through the use of fusion.

Table 7-2 Classifier Performance on Testing Data

Partition Name	P_{cc}	P_{dec}	P_{cc}	P_{dec}	P_{cc}	P_{dec}	P_{cc}	P_{dec}
Tst1-1	0.7922	0.90381						
Tst2-1	0.79095	0.79726						
Tst2-2	0.79381	0.75705	0.81229	0.94424				
Tst2-1st	0.91704	0.73964						
Tst2-2nd	0.43967	0.7954	0.79694	0.94942	0.88116	0.992		
Tst4-1	0.72369	0.80949						
Tst4-2	0.74261	0.85531						
Tst4-3	0.7574	0.83313						
Tst4-4	0.63639	0.75311	0.82266	0.9971				
Tst4-1st	0	0						
Tst4-2nd	0.88051	0.80846						
Tst4-3rd	0.65241	0.76575						
Tst4-4th	0	0	0.83239	0.94983	0.882	0.999		
Tst8-1	0.63722	0.81882						
Tst8-2	0.58947	0.78317						
Tst8-3	0.59995	0.75808						
Tst8-4	0.51747	0.65858						
Tst8-5	0.49859	0.80618						
Tst8-6	0.50673	0.72367						
Tst8-7	0.51172	0.7073						
Tst8-8	0.53631	0.64511	0.77047	0.99979				
Tst8-1st	0	0						
Tst8-2nd	0	0						
Tst8-3rd	0.66444	0.68138						
Tst8-4th	0.81829	0.51451						
Tst8-5th	0.70485	0.41439						
Tst8-6th	0.37494	0.82546						
Tst8-7th	0	0						
Tst8-8th	0	0	0.75351	0.97388	0.84906	0.999	0.923	0.999

An analysis was conducted to ascertain the effect of various parameters on the performance (P_{cc}) of the various classifiers. The first area explored was the effect of the number of training signals. Figure 7-2 shows the results. Training was consistent while testing was not. There was no obvious trend. As the number of training signals increases, there is a higher probability that a test signal will match. Obviously if there are few training signals it will be harder for the test signals to match a training signal. Logically it would seem that there would be an optimal point between too few and too many training signals. The results could be influenced by the partitioning scheme which may account for the inconsistent results.

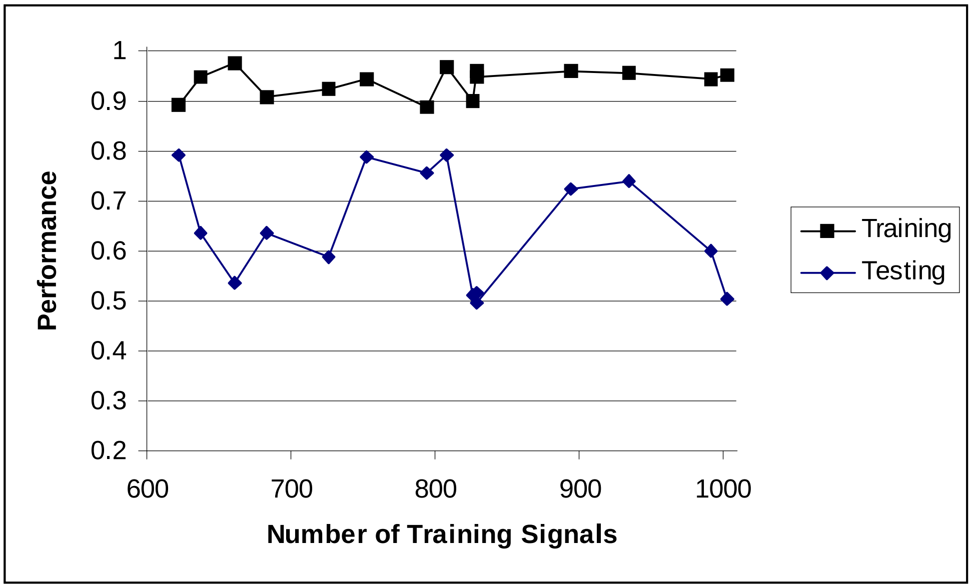


Figure 7-2 Effect of Number of Training Signals on Performance

Another plot was generated to determine if the size of the core had any effect on the classifier performance. This is shown in Figure 7-3. Here it is seen that the size of the core does influence the classifier performance. Recall that a core attribute is the only attribute capable of distinguishing between at least two signals in the training set. This means that there is no redundancy. If that attribute is eliminated through fuzzification then incorrect classification will result. Basically as the core increases the classifier is more dependant on single attributes to do classification. As a result, the more attributes in the core the worse the performance.

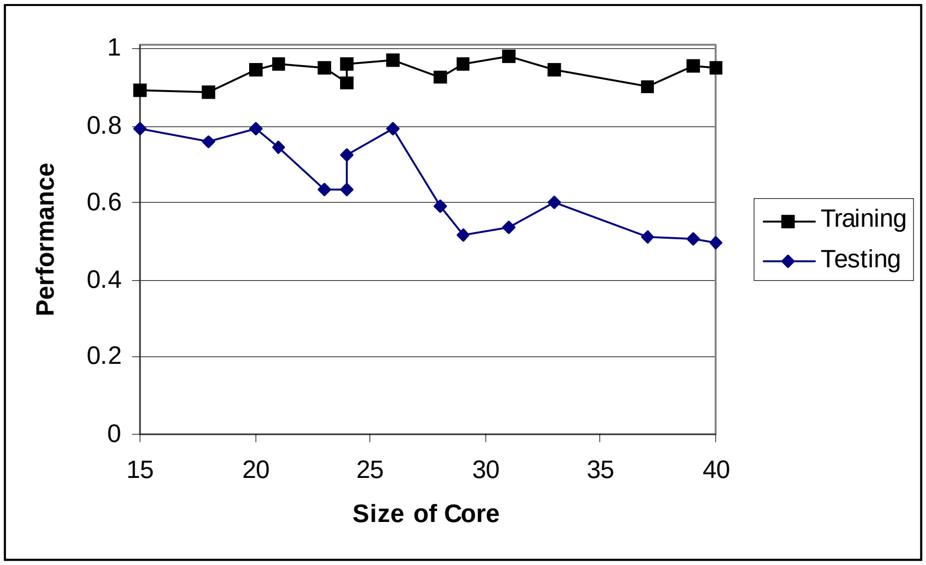


Figure 7-3 Effect of Size of Core on Performance

Another plot was made to determine the effect of minimum reduct size on classifier performance. The results are presented in Figure 7-4. It was conjectured at the beginning of this research that smaller reducts (fewer attributes) would result in better classification. This is because it is easier to match a few attributes than many attributes. The plot bears this out and confirms the conjecture.

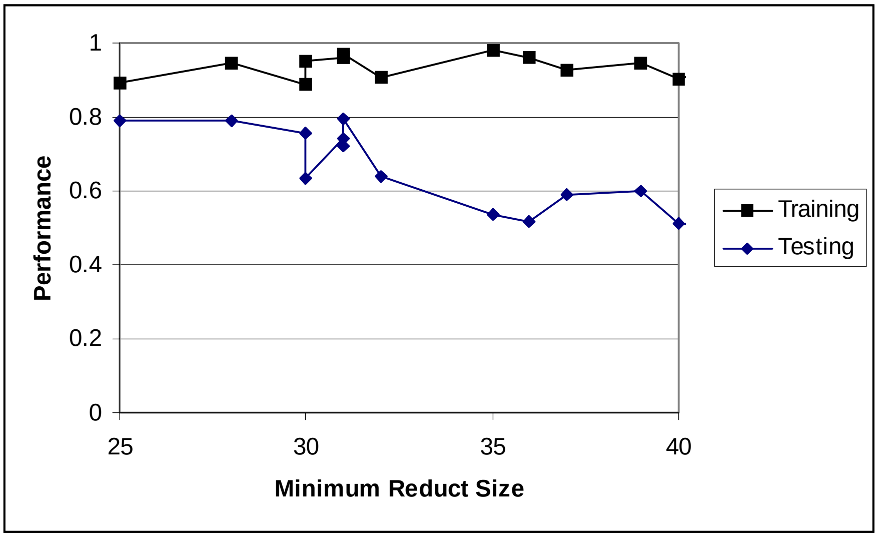


Figure 7-4 Effect of Minimum Reduct Size on Performance

Another question was whether more reducts would result in better classification. This is the idea of many advisors being better than a few. Figure 7-5 shows the results of this analysis. There is an apparent effect that too many reducts is not good. Recall that multiple reducts result when several attributes are found to reduce the same number of ambiguities in the training set. This means that all are roughly equivalent. The reducts generated instead of being independent of each other are approximately the same. When these are fused together they may all have had fairly poor performance on the training set and result in poor performance on the test set.

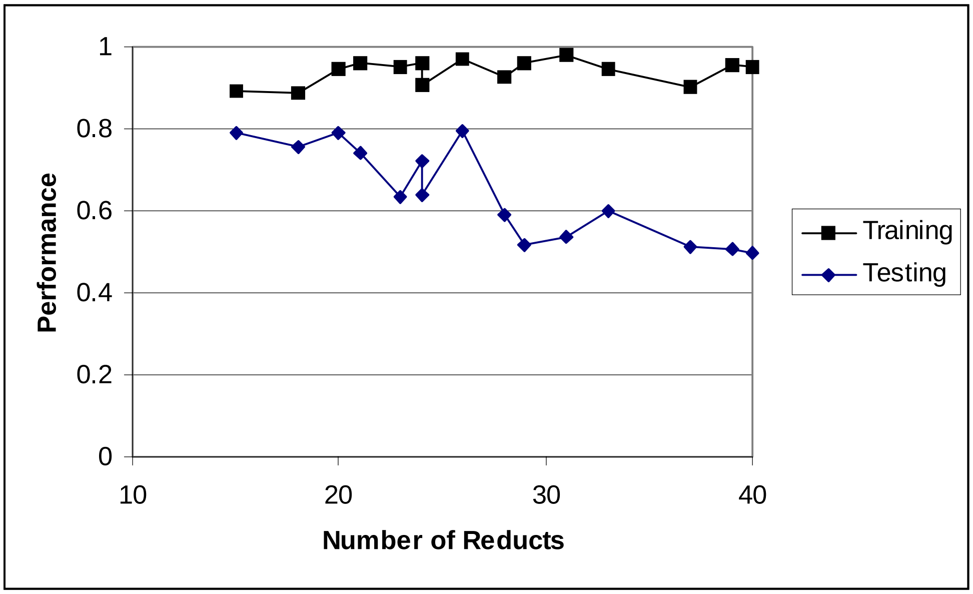


Figure 7-5 Effect of Number of Reducts on Performance

Another area explored was whether there was a relationship between the size of the reducts and the number of reducts. Figure 7-6 confirms that there appears to be a connection. This might be expected because as the number of attributes increases the likelihood that more attributes which produce the same number of ambiguities would be found, thus increasing the number of reducts. As the number of reducts increases and the reduct size increases, as found previously, the performance decreases. It should be noted that the largest reducts and the largest core sizes occur in the interleave partitioning. This may indicated that this partitioning is not particularly good at classification (by itself). Note that when fused with block partitioning of the same size that there is significant improvement in classification performance.

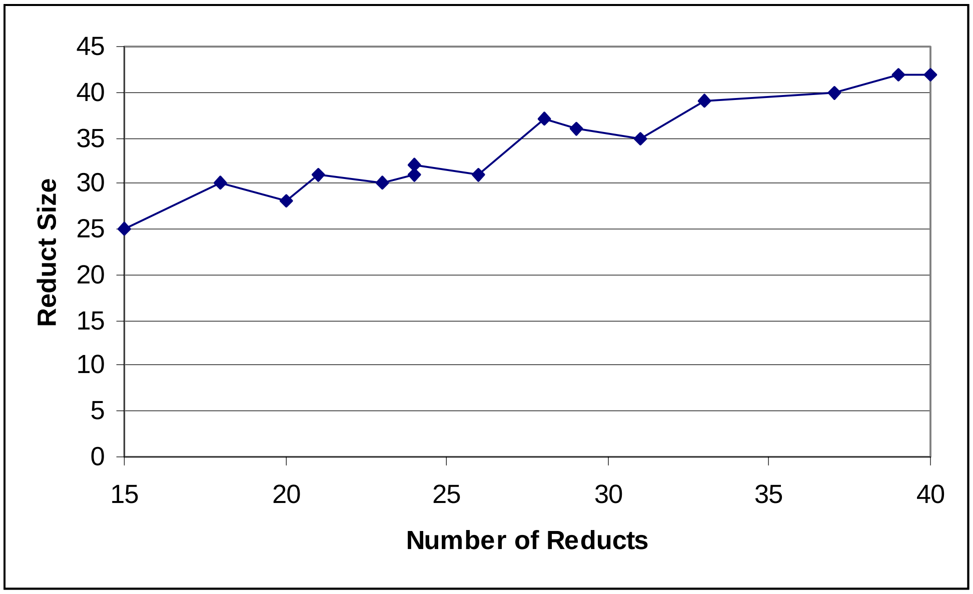


Figure 7-6 Relationship of Number of Reducts to Reduct Size

7.2 Fuzzification Results

The process of fuzzification was described in Section 4.6. Fuzzification was introduced to handle uncertainty regarding data points that are close to the entropy labeling division point. Figure 7-7 shows the effect of different size fuzz factors on classifier performance. Since it only makes sense to use fuzzification on the test data that is what is shown. There is a clear maximum occurring at the 10% point resulting in a 4 percentage point increase in performance. This means that making a buffer or “don’t care” zone of ± 10% of the minimum distance between the dividing point and the terminus values of the attribute yields the best performance. This is interesting because a significant number of attributes are removed but the performance increases. Even though attributes are eliminated others remain which perform the classification task. Apparently these are sufficient when fused with other classifiers to do a good job of classification.

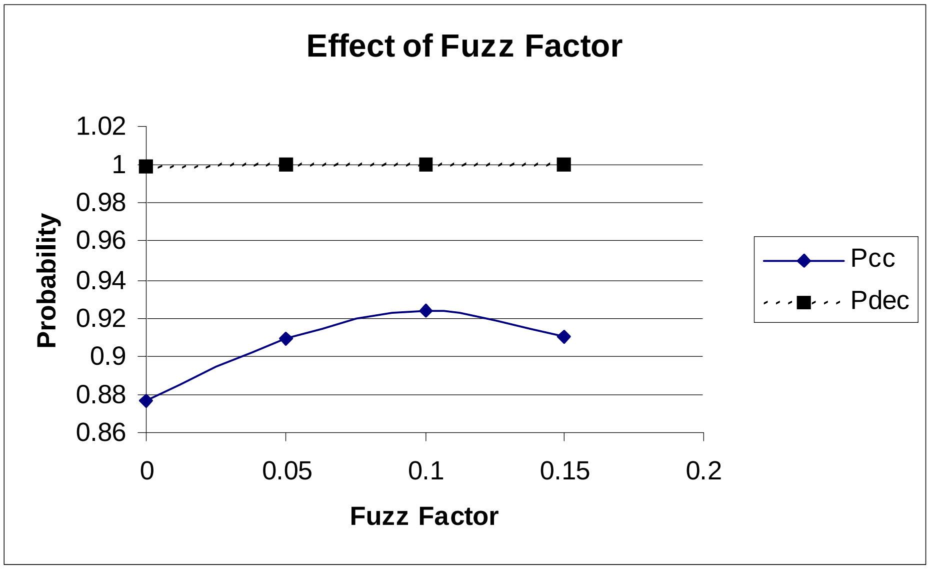


Figure 7-7 Effect of Fuzz Factor on P_{cc} and P_{dec}

7.3 Comparison with Quadratic Classifier

The quadratic classifier is a very popular means of performing the HRR ATR function. A quadratic classifier was trained on the same training data used for this research. It was then tested against the test set. The results, a confusion matrix, are represented in Table 7-3 which has a P_{cc} of 0.8812. This is five percentage points lower than the rough set classifier. On other data with an unknown target class the performance of the quadratic classifier would be much lower. This is due to the fact that the quadratic classifier will classify every signal no matter how close the match. This could be changed through the use of thresholding. The rough set classifier's performance (P_{cc}) would probably not degrade as much as a quadratic classifier because the rough set classifier does not classify a signal if there is no match. The P_{dec} would decrease in this case.

Table 7-3 Quadratic Classifier Confusion Matrix - Test Data

	Target 1	**Target 2**	**Target 3**	**Target 4**	**Target 5**	**Target 6**
Target 1	0.8236	0	0.0112	0.0062	0.0137	0.1456
Target 2	0	0.7738	0.0012	0.2090	0	0.0162
Target 3	0.0025	0.0012	0.9318	0.0547	0.0087	0
Target 4	0	0.0921	0.0037	0.9019	0.0012	0.0012
Target 5	0.0037	0.0025	0.0311	0.0187	0.9430	0.0012
Target 6	0.0149	0	0.0124	0.0485	0.0112	0.9131

7.4 Summary

In this chapter the results of using a rough set classifier on testing data were presented. The classifier was found to have acceptable performance when all the focused reducts from the power information set were fused together. Relationships between the size of the core and reducts was explored. In general, reducts with more attributes and more reducts were found to have lower performance. Some of this was due to the particular partition on which these were based having low target information content.

The use of fuzzification on the test data resulted in a four percentage point increase in accuracy of the classifier. This increase in performance was due to the uncertainty present in some of the range bins when the value was too close to the labeling point. Range bins are only used in the classification process when it is clear which is the correct label to use.

The rough set classifier was found to be better than the most popular method in widespread use today, the quadratic classifier. If unknown targets were included it is likely that this performance difference would increase.

8 Conclusions and Recommendations for Future Studies

8.1 Conclusions

This research met the research objectives as described in Section 1.2 of this dissertation. Specifically, a workable robust classification methodology was developed. The approach used wavelets as a feature generator to enrich the feature set from which a classifier could be developed. The most important features were determined through the use of information entropy. Using these features and rough set theory a multiplicity of minimal classifiers was found. These classifiers were fused together using a formula that was based on the number of classifiers “voting” for a target and their performance to properly weight their effect on the answer. Once a classifier is developed the resulting system is computationally fast and requires little memory for deployment.

Since rough set theory was chosen as a method of finding individual classifiers based on a subset of the attributes, a method had to be developed and the theory itself extended to allow use on larger problems. A method for determining minimal reducts which is polynomial in time complexity greatly increased the number of attributes that could be considered. In fact, the problem used in this research is about two orders of magnitude larger than what has been found in the literature.

These advances produced a software system as evidenced by the output contained in Table A-1 through Table A-29. These tables are included to provide completeness to the results of the research and to illustrate the detail of output from the computer program. Whereas the information contained in the appendix is summarized in the dissertation body the appendix provides additional information which can be used to develop further analysis of this methodology.

This research addressed the serious concerns as quoted in Section 1.3 from the father of rough sets, Zdzislaw Pawlak Specifically, a means to handle large data sets, efficient generation of rules based on quantitative attributes, and an effective discretization method for quantitative attributes.

8.2 Original Contributions

The following is a list of the original contributions developed as a result of this research:

1. The first original contribution developed in this dissertation was the concept of iterated wavelets. Wavelets and their properties have been well known in the literature and used successfully in pattern recognition. In this research the advantages of iterated wavelets were revealed. It was statistically shown that of the many families of wavelets, no family performed better than another. It was further shown that no wavelet in a family was better than another. Many researchers have used a multilevel wavelet decomposition. An iterated wavelet is different in that after a multilevel wavelet decomposition the most salient wavelet coefficients are chosen and another multilevel decomposition is done. This is repeated. This resulted in an 11 percentage point improvement in overall classifier performance after 6 iterations.

2. The expansion algorithm for the discernibility function is an original contribution. Through the use of this algorithm and the concept of strong equivalence the algorithms for computing the reducts using the discernibility relation were vastly improved.

3. The concept of multi-class entropy used in rough set based classification is an original contribution. Although others have used entropy and some even multi-class entropy the way the entropy is computed to provide binary labeling is original. It is through the use of this labeling and bin selection based upon it that rough set theory can be used for target classification.

4. The concept of focused reducts is original. It is not possible to consider enough range bins to achieve adequate performance without the use of the partitioning providing focused reducts which greatly improve classifier performance. It was combined with the development of a new direction in rough set theory which defined and successfully used the power information system.

5. The means of computing reducts is original. The problem of finding all reducts is a non-polynomial time problem. The developed method for finding near minimal reducts makes it of polynomial time complexity. Also shown was that the reducts found are sufficient to solve the given problem.

6. The method of fusing focused reducts is original. The fusion scheme is instrumental in increasing the performance of a classification scheme based on rough set theory.

7. The method of adding fuzziness to the rough set classifier is original. Since there is some doubt of labeling near the dividing point the idea of ignoring those range bins within a small distance of the division point is unique to this method.

8.3 Publications

The following publications resulted from this work:

8.3.1 Journal Articles:

J. A. Starzyk, D. E. Nelson, and K. Sturtz, “Reduct Generation in Information Systems,” *Bulletin of the International Rough Set Society,* vol. 3, Mar.1999.

J. A. Starzyk, D. E. Nelson, and K. Sturtz, “A Mathematical Foundation for Improved Reduct Generation in Information Systems,” *Knowledge and Information Systems*, vol 2., pp. 131-146, 2000.

D. E. Nelson and J. A. Starzyk, “Iterative Wavelet Transformation and Signal Discrimination for HRR Radar Target Recognition,” *IEEE Journal on Systems, Man, and Cybernetics*, in review.

D. E. Nelson, J. A. Starzyk, and D. D.Ensley, “Wavelet Transformation and Signal Discrimination for HRR Radar Target Recognition,” *Multidimensional Systems and Signal Processing*, in review.

D. E. Nelson, J. A. Starzyk, “Fusion of Focused Rough Set Classifiers,” *Pattern Analysis and Applications*, in review.

8.3.2 Patent:

Object Identification System and Method, Letters Patent of the United States serial number 60/220,768 filed July 21, 2000.

8.3.3 Conferences Papers:

J. A. Starzyk and D. E. Nelson., “Independent Classifiers in Ontogenic Neural Networks for ATR,” *Adaptive Distributed Parallel Computing Symposium,* Fairborn, OH, 1996.

D. E. Nelson and J. A. Starzyk, “Advanced Feature Selection Methodology for Automatic Target Recognition,” *Proc. Southeastern Symposium on System Theory*, Cookeville, TN, 1997.

D. E. Nelson, J. A. Starzyk, and K. Sturtz, “Reduct Generation in Information Systems,” *The 6th Int. Workshop on Rough Sets, Data Mining and Granular Computing RSDMGrC'98,* Oct. 1998.

D. E. Nelson and J. A. Starzyk, “Fusing Marginal Reducts for HRR Target Identification,” *Proc. of the 4th World Multiconference on Systemics, Cybernetics, and Informatics*, Jul. 2000. **(Best Paper Award)**

D. E. Nelson and J. A. Starzyk, “High Range Resolution Radar Signal Classification: A Partitioned Rough Set Approach,” *Proc. of the 33rd Southeastern Symposium on System Theory*, Athens, OH, Mar. 2001.

D. E. Nelson and J. A. Starzyk., “High Range Resolution Radar Extensions to Rough Set Theory for Automatic Target Recognition,” *Proc. of the 5th World Multiconference on Systemics, Cybernetics, and Informics*, Orlando, FL, Jul. 2001.

8.4 Recommended Future Studies

In any research there is always more research that can and should be done. The following sections describe areas which still need to be verified and methods found for overcoming known problems.

8.4.1 Sensitivity to Registration

The HRR signals used in this research were synthetically generated by a computer program known as XPATCH. The images were all perfectly registered. That is, all signals began at exactly the same point. This makes classification easier. If the signals are mis-registered then the new signals’ range bin 1 could be range bin 0 or range bin 2 if the signal was mis-registered by one bin. It is possible for the image to be mis-registered by more than that. Because this research used focused reducts with a block and an interleave scheme it is felt that it should be able to deal with small amounts of mis-registration. With the use of binary labeling the effect of mis-registration should also be somewhat mitigated.

Although mis-registration could be a serious shortcoming of this method there are ways to handle the registration problem. Statistical methods such as proposed by Mitchell [22] could handle this problem. A pure statistical approach matching centroids could also be used.

8.4.2 Different Data Sets

This effort used synthetic data. There is never enough measured data to satisfy ATR researchers. Also in world conflict we can seldom convince our adversary to provide us with a copy of their military hardware to get sufficient measured data. Therefore it is likely that it would be necessary to train using synthetic and then test using measured data. This should be explored. In the interest of keeping this research unclassified this was not accomplished.

In this dissertation 25% of the data was used for training and 75% for testing. Frequently, the larger amount of data is used for training and the smaller for testing. The effect of the size of these two data sets should be explored. The training data was selected randomly and the rest of the data was used for testing. Since random selection was used there is no guarantee that there were not areas with large gaps in the data. Uniform sampling should also be accomplished to determine the effect this may have on performance.

Since the data was generated synthetically there was no noise component. This method should be tested to determine the effect of noise on classification performance. In a real world environment one could expect to encounter noisy signals. It is believed that because of the binary labeling and the effect that wavelets have on the signals that this method should be robust to noise.

The data used in this research was air-to-air targets. This has the effect that there is no clutter in the background to confuse the ATR system. The military has strong interest in identifying ground targets from the air. Future research could determine how effective this method would be against these ground targets. It is possible that methods would need to be developed to eliminate ground clutter before performing identification.

Other researchers such as Mitchell [22] have used a multi-look approach to increase classifier performance. This should be explored as a means to increase the performance of the rough set classifier developed in this research.

8.4.3 Number of Range Bins Selected for Reduct Determination

As discussed in Section 6.4.1, the number of range bins to be considered for calculation of the reducts should probably be based on the distribution of the information entropy index. When the curve appears to be very flat, more range bins should be used. When the curve is very steep (large negative slope), fewer range bins should be considered. It is recommended that a method be developed based on covering a percentage of the area under the information entropy index curve. The sensitivity of classification based on this could be an additional study.

Bibliography

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image Coding Using Wavelet Transform,” *IEEE Transactions on Image Processing*, vol. 1, pp. 205-220, 1992.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press, 1995.

[3] I. Borg and P. Groenen, *Modern Multidimensional Scaling*, Berlin: Springer-Verlag, 1997.

[4] T. M. Cover and J. M. Van Capenhout, “On the Possible Orderings in Measurement Selection Process,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 7, pp.657-661, Sep. 1977.

[5] A. Czyzewski, “Speaker Independent Recognition of Digits Based on Soft Computing Methods,” Bulletin of the Polish Academy of Sciences Technical Sciences, vol. 44, Mar.1996.

[6] H. Demuth and M. Beale, *Neural Network Toolbox User’s Guide*, The Math Works, Inc., 1998.

[7] A. J. Devaney and B. Hisconmez, “Wavelet Signal Processing for Radar Target Identification a Scale Sequential Approach,” *SPIE Wavelet Applications*, vol. 2242, pp. 389-399, 1994.

[8] A. J. Devaney, R. Raghavan, H. Lev-Ari., E. Manolakos, and M. Kokar, “Automatic Target Detection and Recognition: A Wavelet Based Approach,” Northeastern University, Defense Technical Information Center, Tech. Rep. AD-A329696, 1997.

[9] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*, New York: John Wiley & Sons, 1996.

[10] K. Etemad and R. Chellapa, “Separability-Based Multiscale Basis Selection and Feature Extraction for Signal and Image Classification,” *IEEE Trans. on Image Processing*, vol. 7, pp. 1453-1465, 1998.

[11] A. Famili, S. Wei-Min, R. Weber, and E. Simoudis, “Data Preprocessing and Intelligent Data Analysis,” *Intelligent Data Analysis*, vol. 1, Jan. 1997.

[12] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., San Diego, CA: Academic Press, 1990.

[13] H. Hotelling, “Analysis of a Complex of StatisticalVariables into Principal Components,” *Journal of Educational Psychology*, vol. 24, pp. 417-441, 1993.

[14] M. Ichino, “Optimum Feature Selection for Decision Functions,” *Systems and Computers in Japan*, vol. 21, Jan. 1990.

[15] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical Pattern Recognition: A Review,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4-37, Jan. 2000.

[16] R. Kohavi and B. Frasca, “Useful Feature Subsets and Rough Set Reducts,” *3rd Int. Workshop on Rough Sets and Soft Computing*, 1994.

[17] T. Kohonen, *Self-Organizing Maps*, *Springer Series in Information Sciences*, vol. 30, Berlin: 1995.

[18] J. Lu, V. R. Algazi, and R. R. Estes, Jr., “Comparative Study of Wavelet Image Coders,” *Optical Engineering*, vol. 35, pp. 2605-2619, 1996.

[19] S. Mallat, “Zero-Crossings of a Wavelet Transform,” *IEEE Transactions on Information Theory*, vol. 37, pp. 1019-1033, 1991.

[20] S. Mallat and Sifen Zhong, “Characterization of Signals from Multiscale Edges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710-732, 1992.

[21] I. Miller and J. E. Fruend, *Probability and Statistics for Engineers*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1965.

[22] R. A. Mitchell, “Hybrid Statistical Recognition Algorithm for Aircraft Identification,” Dissertation, University of Dayton, Dec. 1997.

[23] R. A. Mitchell and J. J. Westerkamp, “High Range Resolution Target Identification Using a Statistical Feature Based Classifier With Feature Level Fusion,” *Automatic Target Recognition Working Group*, Oct. 1997.

[24] R. A. Mitchell and J. J. Westerkamp, “A Statistical Feature Based Classifier for Robust High Range Resolution Radar Target Identification,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, Mar. 1999.

[25] M. Misiti, Y. Nisiti, G. Oppenheim, and J. Poggi, *Wavelet Toolbox User’s Guide*, The MathWorks, Inc., Natick, MA: 1996.

[26] D. E. Nelson and J. A. Starzyk, “Advanced Feature Selection Methodology for Automatic Target Recognition,” *29th Southeastern Symposium on System Theory*, 1997.

[27] H. Niemann, “Linear and Nonlinear Mappings of Patterns,” *Pattern Recognition*, vol. 12, pp. 83-87, 1980.

[28] M. Partridge and M. Jabre, “Robust Principal Component Analysis,” *Neural Networks for Signal Processing X, 2000, Proceedings of the 2000 IEE Processing Society Workshop*, vol. 1, 2000, pp. 289-298.

[29] Z. Pawlak, *Rough Sets Theoretical Aspects of Reasoning About Data*, Dordrecht, The Netherlands: Kluwer Academic Publishers, 1991.

[30] Z. Pawlak, “AI and Intelligent Industrial Applications: The Rough Set Perspective,” *Cybernetics and Systems: An International Journal*, vol. 31, Apr. 2000.

[31] W. Pedrycz, “Shadowed Sets: Bridging Fuzzy and Rough Sets,” *Rough Fuzzy Hybridization A New Trend in Decision-Making*, S. K. Pal and A. Skowron, Eds., Singapore: Springer, 1999, pp. 179-199.

[32] L. Polkowski and A. Skowron, *Rough Sets in Knowledge Discovery 1: Methods and Applications*, New York: Springer-Verlag, 1998.

[33] W. H. Press, S. A. Teukolosky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge: Cambridge University Press, 2nd ed., 1992.

[34] J. W. Prueher, “Information Age Overload: More Data Does Not Mean Superior Judgement,” *Defense News*, pp. 23, Nov. 30 - Dec. 6, 1998.

[35] Y. N. Rao and J. C. Principe, “A Fast On-Line Algorithm for PCA and its Convergence Characteristics,” *Neural Networks for Signal Processing X, 2000, Proceedings of the 2000 IEE Processing Society Workshop*, vol. 1, 2000, pp. 299-307.

[36] E. Rich, *Artificial Intelligence*, New York: McGraw Hill Book Company, 1983.

[37] J. W. Sammon Jr., “A Nonlinear Mapping for Data Structure Analysis,” *IEEE Transactions on Computers,* vol. 18, pp. 401-409, 1969.

[38] G. Shafer, *A Mathematical Theory of Evidence*, Princeton, N. J: Princeton University Press, 1976.

[39] A. Skowron and S. H. Nguyen, “Quantization of RealValued Attributes: Rough Sets and Boolean Reasoning Approach,” Institute of Computer Science Research Rep., Warsaw University of Technology, Nov. 1995.

[40] A. Skowron and C. Rauszer, “The Discernibility Matrices and Functions in Information Systems,” in *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, Dordrecht, The Netherlands: Kluwer, 1992, pp. 331-362.

[41] K. Slowinski, J. Stefanowski, and W. Twardosz, “Rough Set Theory and Rule Induction Techniques for Discovery of Attribute Dependencies in Experience with Multiple Injured Patients,” Bulletin of the Polish Academy of Sciences, vol. 46, Feb. 1998.

[42] J. A. Starzyk, “Evolutionary Feature Extraction for SAR Air to Ground Moving Target Recognition - Statistical Approach,” Sverdrup Technology, Inc. Advanced Systems Group Rep., 1998.

[43] J. A. Starzyk and D. E. Nelson, “Independent Classifiers in Ontogenic Neural Networks for ATR,” *Adaptive Distributive Parallel Computing Symposium*, 1996.

[44] J. A. Starzyk, D. E. Nelson, and K. Sturtz, “Reduct Generation in Information Systems,” *Bulletin of the International Rough Set Society*, vol. 3, Mar. 1999.

[45] J. A. Starzyk, D. E. Nelson, and K. Sturtz, “A Mathematical Foundation for Improved Reduct Generation in Information Systems,” *Knowledge and Information Systems*, vol 2., pp. 131-146, 2000.

[46] C. Stirman and A. Nachman, “Applications of Wavelets to Radar Data Processing,” Defense Technical Information Center, Tech. Rep. AD-A239297, 1991.

[47] C. Stirman and A. Nachman, “Applications of Wavelets to Automatic Target Recognition,” Defense Technical Information Center, Tech. Rep. AD-A294854, 1995.

[48] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA: 1996.

[49] H. H. Szu, “Review of Wavelet Transforms for Pattern Recognition,” *Wavelet Applications III, Proceedings of SPIE*, vol. 2762, pp. 2-22, 1996.

[50] S. Verdu, “Fifty Years of Shannon Theory,” *IEEE Transactions on Information Theory*, vol. 44, Oct. 1998.

[51] A. Wakulicz-Deja andP. Paszek, “Diagnose Progressive Encephalography Applying the Rough Set Theory,” International Journal of Medical Informatics, vol. 46, 1997.

Appendix

This appendix contains the details of the reducts for each of the partitions.

Table A-1 Partition Trn1-1
Range bins considered = 800 798 671 541 796 794 667 394 275 164 165 772 517 774 647 776 923 643 922 409 901 770 904 73 925 928 814 687 816 72 899 557 943 942 306 516 410 810 773 898 683 812 274 646 775 273 927 926 307 393
Core size = 15 Core consists of original bin numbers: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393
Reduct size = 25 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 923 926 773 273 925 683 794 774 798 770
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 925 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 776 928 796 164 165 901 770 942

Table A-1 Partition Trn1-1 (Continued)
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 776 928 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 557 798 901 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 557 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 770 942
Reduct size = 26 Reduct consists of bins: 394 275 772 643 409 73 72 943 306 410 810 898 274 307 393 922 927 516 647 928 796 164 165 901 770 942

Table A-2 Partition Trn2-1
Range bins considered = 138 83 287 350 352 37 221 410 411 324 259 346 283 197 348 322 389 392 263 326 328 154 368 387 303 366 325 415 414 237 262 327 386 362 27 299 81 196 364 330 82 267 332 336 205 271 334 34 115 36
Core size = 20 Core consists of original bin numbers: 138 83 37 410 411 259 389 326 154 325 414 327 27 81 196 82 334 34 115 36
Reduct size = 28 Reduct consists of bins: 138 83 37 410 411 259 389 326 154 325 414 327 27 81 196 82 334 34 115 36 368 350 267 362 387 324 346 386
Reduct size = 28 Reduct consists of bins: 138 83 37 410 411 259 389 326 154 325 414 327 27 81 196 82 334 34 115 36 368 350 267 362 387 324 346 386

Table A-3 Partition Trn2-2
Range bins considered = 352 350 287 221 346 283 348 153 36 324 389 387 326 386 392 263 328 197 83 82 138 259 37 325 416 413 262 327 196 322 137 411 410 81 430 431 154 114 323 365 367 302 21 236 258 368 303 366 429 432
Core size = 26 Core consists of original bin numbers: 153 36 324 389 326 386 328 197 83 82 138 259 37 325 196 322 137 411 410 81 154 114 323 21 258 429
Reduct size = 31 Reduct consists of bins: 153 36 324 389 326 386 328 197 83 82 138 259 37 325 196 322 137 411 410 81 154 114 323 21 258 429 416 368 430 346 236

Table A-3 Partition Trn2-2 (Continued)
Reduct size = 31 Reduct consists of bins: 153 36 324 389 326 386 328 197 83 82 138 259 37 325 196 322 137 411 410 81 154 114 323 21 258 429 416 368 430 346 236
Reduct size = 31 Reduct consists of bins: 153 36 324 389 326 386 328 197 83 82 138 259 37 325 196 322 137 411 410 81 154 114 323 21 258 429 413 368 430 346 236
Reduct size = 31 Reduct consists of bins: 153 36 324 389 326 386 328 197 83 82 138 259 37 325 196 322 137 411 410 81 154 114 323 21 258 429 413 368 430 346 236
Reduct size = 31 Reduct consists of bins: 153 36 324 389 326 386 328 197 83 82 138 259 37 325 196 322 137 411 410 81 154 114 323 21 258 429 413 303 430 346 236

Table A-4 Partition Trn2-1st
Range bins considered = 206 267 271 387 157 324 388 117 187 260 87 139 46 43 49 280 53 215 398 334 397 86 89 408 407 344 263 198 336 400 399 140 123 42 259 142 119 56 143 93 92 272 223 121 118 85 91 342 406 405
Core size = 16 Core consists of original bin numbers: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 93 408 89 398 336 223 142 342 387 334 42

Table A-4 Partition Trn2-1st (Continued)
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 408 223 397 336 49 206 387 42 342 157 142
Reduct size = 28 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 260 400 49 342 206 42 387 157 142
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 336 49 206 387 42 342 157 142
Reduct size = 28 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 334 157 336 89 342 398 206 42 142 387 92
Reduct size = 28 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 260 400 49 342 206 42 387 157 142
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 336 49 206 387 42 342 157 142
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 336 49 206 387 42 342 157 142
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 336 49 206 387 42 342 157 92

Table A-4 Partition Trn2-1st (Continued)
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 336 49 206 387 42 406 157 142
Reduct size = 27 Reduct consists of bins: 117 187 87 139 46 43 53 215 86 140 123 119 56 143 121 91 344 223 397 336 49 206 387 42 342 157 142

Table A-5 Partition Trn2-2nd
Range bins considered = 400 335 399 129 269 333 398 397 201 68 69 65 95 62 92 61 144 56 143 200 59 93 57 55 58 260 63 94 96 60 322 64 142 52 199 141 49 89 90 91 53 48 54 50 140 47 51 88 261 198
Core size = 14 Core consists of original bin numbers: 335 399 129 397 201 68 69 65 61 142 141 47 51 261
Reduct size = 30 Reduct consists of bins: 335 399 129 397 201 68 69 65 61 142 141 47 51 261 200 64 333 143 49 89 400 92 269 398 95 62 144 56 59 58
Reduct size = 30 Reduct consists of bins: 335 399 129 397 201 68 69 65 61 142 141 47 51 261 143 64 54 333 49 89 400 95 269 398 62 92 144 56 59 58
Reduct size = 30 Reduct consists of bins: 335 399 129 397 201 68 69 65 61 142 141 47 51 261 143 64 91 333 49 89 400 95 269 398 62 92 144 56 59 58

Table A-6 Partition Trn4-1
Range bins considered = 42 19 142 111 144 77 58 140 168 107 138 165 133 102 132 99 186 135 187 169 14 172 55 69 160 130 127 158 189 192 174 175 13 68 93 163 153 122 155 162 134 78 103 41 136 123 156 154 92 178
Core size = 24 Core consists of original bin numbers: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 186 169 168 142 132 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 186 169 168 142 132 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 169 168 142 132 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 168 142 132 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 168 142 132 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122

Table A-6 Partition Trn4-1 (Continued)
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 158 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 155

Table A-6 Partition Trn4-1 (Continued)
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 136 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 142 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 158 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122

Table A-6 Partition Trn4-1 (Continued)
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 155
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 136 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 155
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 127 122

Table A-6 Partition Trn4-1 (Continued)
Reduct size = 31 Reduct consists of bins: 42 19 58 107 138 102 135 14 55 130 189 192 174 175 13 68 93 163 153 78 41 156 92 178 187 172 134 111 165 158 122

Table A-7 Partition Trn4-2
Range bins considered = 19 42 127 158 160 93 133 58 165 168 102 123 156 135 154 91 83 68 54 99 187 186 132 130 163 162 69 11 40 59 75 78 12 147 114 142 144 111 145 134 103 94 136 38 77 86 138 140 107 67
Core size = 21 Core consists of original bin numbers: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 127 114 91 187 186 102 142 158 165 103
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 127 114 91 187 186 102 142 165 156 103
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 127 114 91 187 186 102 142 158 165 103
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 158 147 187 54 75 142 165 102 136 135

Table A-7 Partition Trn4-2 (Continued)
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 158 147 186 54 156 136 165 19 102 144
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 158 147 186 54 156 136 165 19 102 142
Reduct size = 31 Reduct consists of bins: 42 133 58 168 68 99 132 130 162 69 11 40 59 78 12 94 38 77 86 138 67 158 147 186 54 156 136 165 19 102 142

Table A-8 Partition Trn4-3
Range bins considered = 127 160 158 132 93 103 136 134 154 123 156 69 192 189 18 57 91 187 186 54 163 99 83 191 162 190 177 180 17 41 165 168 130 11 68 152 119 150 170 159 137 126 157 85 171 106 148 115 146 182
Core size = 18 Core consists of original bin numbers: 132 136 57 163 190 17 41 165 168 130 11 68 137 157 85 171 106 182
Reduct size = 30 Reduct consists of bins: 132 136 57 163 190 17 41 165 168 130 11 68 137 157 85 171 106 182 187 177 154 192 18 127 54 152 93 103 148 189
Reduct size = 31 Reduct consists of bins: 132 136 57 163 190 17 41 165 168 130 11 68 137 157 85 171 106 182 186 177 154 192 18 127 54 152 103 148 93 189 99

Table A-8 Partition Trn4-3 (Continued)
Reduct size = 31 Reduct consists of bins: 132 136 57 163 190 17 41 165 168 130 11 68 137 157 85 171 106 182 186 180 154 192 18 127 54 152 103 148 93 189 99

Table A-9 Partition Trn4-4
Range bins considered = 57 18 93 132 136 134 103 160 69 123 156 127 154 158 99 41 162 163 152 119 165 150 85 168 148 39 115 146 192 189 130 170 171 68 14 138 107 140 17 106 137 139 77 142 111 76 144 133 131 102
Core size = 23 Core consists of original bin numbers: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131
Reduct size = 30 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 152 133 146 171 160 142 148
Reduct size = 30 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 152 133 146 171 160 142 148
Reduct size = 30 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 152 133 146 171 160 142 148
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 152 133 171 148 160 163 85 106

Table A-9 Partition Trn4-4 (Continued)
Reduct size = 30 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 133 146 171 160 142 148
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 146 39 142 160 148 170
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 160 142 170
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 160 142 170
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 160 142 170
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 160 111 170
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 160 142 170
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 156 142 170

Table A-9 Partition Trn4-4 (Continued)
Reduct size = 31 Reduct consists of bins: 57 18 132 136 158 99 41 162 165 168 192 189 130 68 14 138 17 137 139 77 76 144 131 119 102 148 39 85 156 111 170

Table A-10 Partition Trn4-1st
Range bins considered = 48 32 161 72 100 31 129 162 132 134 30 47 130 165 97 98 29 163 35 104 169 80 166 66 68 6 108 65 36 138 28 34 7 33 4 2 37 46 3 67 10 1 11 99 9 40 5 15 39
Core size = 23 Core consists of original bin numbers: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39
Reduct size = 33 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 162 3 66 31 10 98 129 134 68 138
Reduct size = 33 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 162 3 66 31 10 98 129 68 65 138
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 162 134 31 34 10 98 66 129 68 65 138
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 98 35 67 97 68 65 37

Table A-10 Partition Trn4-1st (Continued)
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 98 35 67 97 65 37 40
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 98 35 67 97 68 65 37
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 98 35 67 97 68 65 37
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 35 67 40 97 65 37 15
Reduct size = 33 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 34 97 35 68 15 67 65 37
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 35 68 67 97 65 37 15
Reduct size = 34 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 138 34 35 67 40 97 65 37 15
Reduct size = 33 Reduct consists of bins: 32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39 134 130 34 97 35 68 15 67 65 37

Table A-10 Partition Trn4-1st (Continued)

Reduct size = 34
Reduct consists of bins:
32 132 30 47 165 29 163 104 169 80 166 108 28 7 33 4 2 46 11 99 9 5 39
134 130 138 34 35 68 67 97 65 37 15

Table A-11 Partition Trn4-2nd

Range bins considered = 102 133 135 162 77 53 91 130 39 67 11 14 17 140 107 167 21
38 41 172 131 98 168 68 59 129 10 70 55 24 45 71 44 136 111 57 54 37 43 171
85 78 104 26 25 103 92 165 108 157

Core size = 3
Core consists of original bin numbers:
14 167 55

Reduct size = 25
Reduct consists of bins:
14 167 55 162 44 133 135 102 21 136 11 77 53 172 45 85 157 140 165 78 54 24
92 171 168

Table A-12 Partition Trn4-3rd

Range bins considered = 168 65 135 167 101 36 33 37 67 131 74 172 163 164 139 8 32
98 48 171 9 31 72 47 102 30 29 75 100 28 46 27 71 34 45 26 25 105 1 44
24 23 141 173 174 70 130 22 43 97

Core size = 25
Core consists of original bin numbers:
101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22
43 97

Table A-12 Partition Trn4-3rd (Continued)
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 48 25 168 72 100 141 30 44
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 48 25 168 72 100 141 30 44
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 48 25 168 72 100 141 28 44
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 168 44 141 30
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 168 44 141 30
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 168 44 141 30
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 65 30 44 168 28 141
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 168 44 141 30

Table A-12 Partition Trn4-3rd (Continued)
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 32 28 65 100 168 44 141
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 168 44 173 30
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 168 44 141 30
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 65 44 135 30 141
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 32 28 65 100 135 44 141
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 65 44 135 30 173
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 72 65 44 135 30 141
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 32 28 65 100 135 44 141

Table A-12 Partition Trn4-3rd (Continued)
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 65 30 44 135 28 173
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 65 30 44 135 28 141
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 65 30 44 135 28 141
Reduct size = 33 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 65 30 44 135 28 173
Reduct size = 32 Reduct consists of bins: 101 36 33 37 67 131 74 172 163 164 8 98 171 9 102 75 34 105 1 23 174 130 22 43 97 31 27 65 28 44 135 173

Table A-13 Partition Trn4-4th
Range bins considered = 161 130 162 97 129 33 65 3 2 34 131 27 133 45 1 71 26 70 25 44 4 24 23 31 100 46 165 72 48 32 28 47 30 66 43 137 102 99 29 35 5 163 22 101 6 21 16 9 106 76 67
Core size = 12 Core consists of original bin numbers: 130 162 27 1 165 137 29 101 6 21 169 76

Table A-13 Partition Trn4-4th (Continued)

Reduct size = 23
Reduct consists of bins:
130 162 27 1 165 137 29 101 6 21 169 76 161 131 106 97 99 33 26 25 65 23 31

Table A-14 Partition Trn8-1

Range bins considered = 10 64 62 47 29 58 60 43 78 79 55 38 53 74 75 20 69 72 73 76 77 80 35 8 50 52 66 67 46 61 63 21 51 59 7 57 42 28 34 9 54 30 39 49 56 27 70 71 6 68

Core size = 24
Core consists of original bin numbers:
62 47 29 78 53 74 75 77 80 35 8 50 52 66 63 21 9 30 49 27 70 71 6 68

Reduct size = 32
Reduct consists of bins:
62 47 29 78 53 74 75 77 80 35 8 50 52 66 63 21 9 30 49 27 70 71 6 68 76 57 69 7 54 60 55 46

Table A-15 Partition Trn8-2

Range bins considered = 10 62 64 47 29 60 43 58 75 74 80 77 53 38 55 27 20 69 72 70 8 6 71 79 78 52 19 35 66 67 50 34 21 59 57 30 42 51 54 39 56 49 63 46 61 9 28 73 76 11

Core size = 28
Core consists of original bin numbers:
10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11

Table A-15 Partition Trn8-2 (Continued)
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 53 34 60 80 63 67 76 38 69
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 53 34 60 80 63 67 76 38 69
Reduct size = 38 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 38 34 60 80 63 76 55 69 72 67
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 49 55 80 69 60 63 72 66 73
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 49 55 77 69 60 63 72 66 73
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 49 55 80 69 60 63 72 66 73
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 49 55 80 69 43 63 72 66 73
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 49 55 77 69 43 63 72 66 73

Table A-15 Partition Trn8-2 (Continued)
Reduct size = 37 Reduct consists of bins: 10 62 29 75 74 27 20 70 8 6 71 79 78 52 19 35 50 59 30 42 51 54 39 56 61 9 28 11 49 55 77 69 43 46 72 66 73

Table A-16 Partition Trn8-3
Range bins considered = 77 80 27 70 6 71 19 74 75 62 47 64 10 29 60 43 58 54 52 56 39 9 53 38 55 49 34 21 66 67 69 72 46 61 63 20 8 35 24 1 16 15 59 57 17 51 42 30 36 2
Core size = 33 Core consists of original bin numbers: 77 80 27 70 6 71 19 74 75 62 64 10 54 56 9 38 55 49 34 21 66 67 46 20 8 35 1 16 15 51 30 36 2
Reduct size = 39 Reduct consists of bins: 77 80 27 70 6 71 19 74 75 62 64 10 54 56 9 38 55 49 34 21 66 67 46 20 8 35 1 16 15 51 30 36 2 59 69 58 39 61 57
Reduct size = 39 Reduct consists of bins: 77 80 27 70 6 71 19 74 75 62 64 10 54 56 9 38 55 49 34 21 66 67 46 20 8 35 1 16 15 51 30 36 2 59 69 58 39 61 42
Reduct size = 39 Reduct consists of bins: 77 80 27 70 6 71 19 74 75 62 64 10 54 56 9 38 55 49 34 21 66 67 46 20 8 35 1 16 15 51 30 36 2 57 69 58 39 61 42
Reduct size = 40 Reduct consists of bins: 77 80 27 70 6 71 19 74 75 62 64 10 54 56 9 38 55 49 34 21 66 67 46 20 8 35 1 16 15 51 30 36 2 57 72 43 47 39 63 42

Table A-16 Partition Trn8-3 (Continued)
Reduct size = 40 Reduct consists of bins: 77 80 27 70 6 71 19 74 75 62 64 10 54 56 9 38 55 49 34 21 66 67 46 20 8 35 1 16 15 51 30 36 2 57 72 43 47 39 61 42

Table A-17 Partition Trn8-4
Range bins considered = 77 80 7 74 75 9 62 47 29 64 39 56 54 66 52 67 58 60 43 21 69 20 72 34 51 53 35 55 38 49 61 63 46 50 28 27 6 10 59 42 57 19 70 71 30 8 68 11 5 22
Core size = 29 Core consists of original bin numbers: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 59 80 61 29 27 47 53
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 59 80 61 29 27 47 53
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 59 80 61 29 27 47 53
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 61 29 27 47 53

Table A-17 Partition Trn8-4 (Continued)
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 61 29 27 47 53
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 29 27 47 38
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 29 27 47 53
Reduct size = 37 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 29 47 53 63 6
Reduct size = 36 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 29 27 47 53
Reduct size = 37 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 53 47 63 6 11
Reduct size = 37 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 38 47 63 6 11
Reduct size = 37 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 53 47 63 6 11

Table A-17 Partition Trn8-4 (Continued)
Reduct size = 37 Reduct consists of bins: 7 74 75 9 62 54 66 52 67 60 43 21 20 72 34 51 55 49 50 28 10 19 70 71 30 8 68 5 22 42 77 46 53 64 63 6 11

Table A-18 Partition Trn8-5
Range bins considered = 69 72 20 7 53 38 67 66 55 35 52 21 51 50 74 57 42 39 59 34 54 56 75 28 49 9 10 27 76 73 63 46 61 80 77 6 47 64 62 29 58 43 60 70 19 71 79 30 78 8
Core size = 40 Core consists of original bin numbers: 69 72 20 7 67 66 35 52 21 51 50 74 57 59 34 54 56 75 28 49 9 10 27 76 73 46 61 80 77 6 62 29 58 60 70 19 71 30 78 8
Reduct size = 42 Reduct consists of bins: 69 72 20 7 67 66 35 52 21 51 50 74 57 59 34 54 56 75 28 49 9 10 27 76 73 46 61 80 77 6 62 29 58 60 70 19 71 30 78 8 64 53

Table A-19 Partition Trn8-6
Range bins considered = 52 35 50 21 67 69 72 66 53 54 38 39 55 56 20 10 36 23 16 27 14 24 15 22 11 13 51 12 1 17 65 30 2 33 9 6 3 34 18 75 74 70 19 8 4 71 68 5 79 78
Core size = 39 Core consists of original bin numbers: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78

Table A-19 Partition Trn8-6 (Continued)
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 39 52 33
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 39 52 33
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 56 52 33
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 56 52 3
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 56 52 33
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 56 52 33
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 56 38 33
Reduct size = 42 Reduct consists of bins: 35 50 21 67 69 72 66 53 54 55 20 10 36 16 27 14 15 22 11 13 51 12 1 65 30 2 9 6 34 75 74 70 19 8 4 71 68 5 78 56 38 33

Table A-20 Partition Trn8-7
Range bins considered = 52 54 39 56 21 35 66 67 50 9 69 58 43 72 60 29 62 64 47 20 7 73 28 10 76 57 51 61 42 59 46 63 75 79 78 74 8 34 65 49 2 24 17 36 53 16 15 1 38 68
Core size = 37 Core consists of original bin numbers: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 73 43 29
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 73 43 29
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 76 43 29
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 76 43 62
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 76 43 29
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 76 43 29

Table A-20 Partition Trn8-7 (Continued)
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 76 60 29
Reduct size = 40 Reduct consists of bins: 52 56 35 66 67 50 9 69 58 72 47 20 7 28 10 57 51 42 59 63 75 79 78 74 8 34 65 49 2 24 36 53 16 15 1 38 68 76 60 29

Table A-21 Partition Trn8-8
Range bins considered = 39 56 54 52 9 21 66 67 35 43 58 64 60 29 62 47 34 51 50 73 76 49 7 69 72 74 75 28 20 77 80 57 42 79 78 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10
Core size = 31 Core consists of original bin numbers: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10
Reduct size = 35 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 42 39 43 54
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 39 43 57 35 64
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 39 58 57 54 35

Table A-21 Partition Trn8-8 (Continued)
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 39 43 57 54 35
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 39 43 57 54 35
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 54 43 57 56 35
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 54 43 57 56 35
Reduct size = 36 Reduct consists of bins: 9 21 66 67 60 34 51 50 76 49 69 75 28 20 80 79 59 8 63 46 61 38 53 55 68 6 19 27 71 70 10 54 58 57 56 35

Table A-22 Partition Trn8-1st
Range bins considered = 36 24 16 2 66 50 49 4 65 33 17 1 18 3 15 34 6 54 19 5 67 7 52 20 56 69 40 70 71 23 14 8 11 22 12 68 35 72 57 13 21 9 10 74 73 59 41 58 61 79
Core size = 2 Core consists of original bin numbers: 40 72

Table A-22 Partition Trn8-1st (Continued)
Reduct size = 37 Reduct consists of bins: 40 72 20 22 4 54 16 15 6 7 10 71 70 79 36 24 2 66 50 49 65 33 17 1 18 3 34 19 5 67 52 69 23 14 8 56 73
Reduct size = 35 Reduct consists of bins: 40 72 4 7 6 54 10 71 15 70 79 36 24 16 2 66 50 49 65 33 17 1 18 3 34 19 5 67 52 69 23 14 8 56 73
Reduct size = 35 Reduct consists of bins: 40 72 4 7 6 54 10 71 15 70 79 36 24 16 2 66 50 49 65 33 17 1 18 3 34 19 5 67 52 69 23 14 8 56 73
Reduct size = 34 Reduct consists of bins: 40 72 4 7 6 54 10 71 70 79 36 24 16 2 66 50 49 65 33 17 1 18 3 34 19 5 67 52 69 23 14 8 56 73
Reduct size = 34 Reduct consists of bins: 40 72 4 7 6 54 10 71 15 70 79 36 24 16 2 66 50 49 65 33 17 1 18 3 19 5 67 52 69 23 14 8 56 73
Reduct size = 14 Reduct consists of bins: 40 72 4 7 6 69 15 12 10 71 70 79 58 68
Reduct size = 12 Reduct consists of bins: 40 72 4 7 6 54 10 70 79 8 56 73
Reduct size = 13 Reduct consists of bins: 40 72 6 12 15 7 34 10 71 70 79 58 68
Reduct size = 12 Reduct consists of bins: 40 72 6 12 34 7 10 71 70 79 58 68

Table A-22 Partition Trn8-1st (Continued)
Reduct size = 13 Reduct consists of bins: 40 72 34 7 5 12 15 10 71 70 58 68 79
Reduct size = 13 Reduct consists of bins: 40 72 6 12 15 7 34 10 71 70 79 58 68
Reduct size = 11 Reduct consists of bins: 40 72 6 12 15 56 73 70 8 21 68
Reduct size = 11 Reduct consists of bins: 40 72 6 12 15 56 73 70 8 21 68
Reduct size = 11 Reduct consists of bins: 40 72 6 12 34 56 73 10 70 9 79

Table A-23 Partition Trn8-2nd
Range bins considered = 24 36 16 65 49 67 15 66 33 50 14 52 40 54 18 4 34 1 5 23 69 17 51 68 19 35 2 3 70 58 22 12 13 56 73 6 53 20 32 39 11 44 38 55 8 74 7 71 10 61
Core size = 5 Core consists of original bin numbers: 69 17 19 58 39
Reduct size = 32 Reduct consists of bins: 69 17 19 58 39 51 61 2 14 55 5 74 40 1 3 68 6 35 54 66 18 32 23 38 70 24 52 65 49 67 16 44

Table A-23 Partition Trn8-2nd (Continued)
Reduct size = 27 Reduct consists of bins: 69 17 19 58 39 35 74 52 53 18 11 34 14 55 54 33 16 40 73 70 56 24 44 1 36 65 4
Reduct size = 28 Reduct consists of bins: 69 17 19 58 39 13 10 35 74 55 11 1 6 4 34 70 52 33 44 23 24 32 65 14 38 36 16 73
Reduct size = 26 Reduct consists of bins: 69 17 19 58 39 13 8 3 68 11 18 1 14 70 52 74 53 56 55 24 23 32 54 73 49 6
Reduct size = 27 Reduct consists of bins: 69 17 19 58 39 13 8 10 20 53 18 54 32 16 68 65 34 38 23 70 56 73 36 35 24 33 44
Reduct size = 27 Reduct consists of bins: 69 17 19 58 39 13 8 10 20 53 18 54 32 16 68 6 65 34 70 23 38 56 33 73 24 36 35
Reduct size = 27 Reduct consists of bins: 69 17 19 58 39 13 8 10 20 53 18 54 32 16 68 6 65 34 70 23 38 56 33 73 24 36 35

Table A-24 Partition Trn8-3rd
Range bins considered = 52 67 68 29 35 47 36 11 22 10 21 14 30 1 15 17 2 33 3 4 18 23 5 48 24 49 19 66 31 6 79 34 64 80 7 39 9 20 12 8 65 55 40 43 38 28 32 42 46 16
Core size = 30 Core consists of original bin numbers: 29 47 36 11 22 10 21 14 30 15 2 23 48 24 66 31 7 39 9 12 8 55 40 43 38 28 32 42 46 16

Table A-24 Partition Trn8-3rd (Continued)
Reduct size = 32 Reduct consists of bins: 29 47 36 11 22 10 21 14 30 15 2 23 48 24 66 31 7 39 9 12 8 55 40 43 38 28 32 42 46 16 35 18
Reduct size = 33 Reduct consists of bins: 29 47 36 11 22 10 21 14 30 15 2 23 48 24 66 31 7 39 9 12 8 55 40 43 38 28 32 42 46 16 52 67 18
Reduct size = 33 Reduct consists of bins: 29 47 36 11 22 10 21 14 30 15 2 23 48 24 66 31 7 39 9 12 8 55 40 43 38 28 32 42 46 16 52 67 18

Table A-25 Partition Trn8-4th
Range bins considered = 37 70 1 5 53 17 52 68 27 65 55 35 25 21 19 8 34 24 41 20 38 13 51 10 36 26 54 9 31 4 16 78 46 40 59 67 48 15 72 74 76 44 11 50 7 49 28 39 61 22
Core size = 10 Core consists of original bin numbers: 17 21 78 67 48 44 11 7 61 22
Reduct size = 28 Reduct consists of bins: 17 21 78 67 48 44 11 7 61 22 5 70 25 38 26 9 8 39 19 41 20 4 76 53 52 68 55 31

Table A-26 Partition Trn8-5th
Range bins considered = 33 68 51 17 20 21 35 66 38 70 8 50 9 52 39 18 53 1 71 49 2 56 54 10 3 37 34 67 55 28 22 7 4 42 11 69 65 12 29 43 41 36 16 25 23 46 58 60 5 24
Core size = 30 Core consists of original bin numbers: 51 17 66 38 8 50 9 18 53 1 71 2 54 10 3 34 67 55 22 4 11 65 12 41 16 25 23 58 24
Reduct size = 35 Reduct consists of bins: 51 17 66 38 8 50 9 18 53 1 71 2 54 10 3 34 67 55 22 4 11 65 12 41 16 25 23 58 24 28 56 69 20 49
Reduct size = 35 Reduct consists of bins: 51 17 66 38 8 50 9 18 53 1 71 2 54 10 3 34 67 55 22 4 11 65 12 41 16 25 23 58 24 28 56 69 20 49
Reduct size = 35 Reduct consists of bins: 51 17 66 38 8 50 9 18 53 1 71 2 54 10 3 34 67 55 22 4 11 65 12 41 16 25 23 58 5 24 7 56 49 33 42

Table A-27 Partition Trn8-6th
Range bins considered = 65 50 35 66 51 36 33 22 67 21 23 17 16 24 12 13 14 68 11 15 9 10 20 8 7 53 34 1 69 37 70 6 2 19 55 41 57 71 73 5 72 59 74 75 49 25 76 3 26 45
Core size = 29 Core consists of original bin numbers: 65 66 67 17 16 12 14 68 11 9 10 8 53 69 37 70 6 2 19 73 5 72 74 75 49 76 3 26 45

Table A-27 Partition Trn8-6th (Continued)
Reduct size = 34 Reduct consists of bins: 65 66 67 17 16 12 14 68 11 9 10 8 53 69 37 70 6 2 19 73 5 72 74 75 49 76 3 26 45 7 23 1 51 33
Reduct size = 34 Reduct consists of bins: 65 66 67 17 16 12 14 68 11 9 10 8 53 69 37 70 6 2 19 73 5 72 74 75 49 76 3 26 45 7 23 55 25 51
Reduct size = 35 Reduct consists of bins: 65 66 67 17 16 12 14 68 11 9 10 8 53 69 37 70 6 2 19 73 5 72 74 75 49 76 3 26 45 1 23 71 51 21 25

Table A-28 Partition Trn8-7th
Range bins considered = 36 24 16 65 2 18 3 50 66 49 67 17 33 1 4 15 69 52 23 51 13 14 19 5 34 54 40 22 12 56 6 53 35 44 7 62 20 8 73 76 11 21 31 68 77 48 58 72 10 9
Core size = 7 Core consists of original bin numbers: 2 51 40 6 77 72 9
Reduct size = 17 Reduct consists of bins: 2 51 40 6 77 72 9 73 53 69 68 19 16 67 56 52 7

Table A-29 Partition Trn8-8th
Range bins considered = 33 73 17 51 35 11 10 68 21 79 18 25 66 50 49 80 20 45 15 78 58 7 3 9 57 2 65 24 16 36 67 8 74 22 1 5 75 62 12 23 14 19 6 4 34 13 71 63 69 53

Table A-29 Partition Trn8-8th (Continued)
Core size = 6 Core consists of original bin numbers: 11 68 18 9 67 53
Reduct size = 12 Reduct consists of bins: 11 68 18 9 67 53 71 10 78 51 25 66

Index of Defintions

NELSON, DALE, EDWARD. Ph.D. June 2001

Electrical Engineering

High Range Resolution Radar Target Classification: A Rough Set Approach (pp. 152)

Director of Dissertation: Dr. Janusz A. Starzyk

High Range Resolution (HRR) radar is one sensor of interest to the military. This sensor collects data which is a range profile of an aircraft, the result of electromagnetic scattering from the target as a function of the line of sight range. Conventional means of developing an ATR system fail to give adequate results and learning techniques must be used. The primary objective of this research was to develop a workable, robust classification methodology using machine learning and data mining techniques. Specifically the approach should: generate features for classification, determine important features, generate multiple classifiers, determine a method of fusing classifiers for robustness, and be computationally appropriate. Rough Set Theory (RST) guarantees that all possible classifiers using a labelled training set will be generated! There is no equivalent statement for statistical pattern recognition. However, generating all classifiers has been shown to be a NP-hard problem. Therefore, this research had a secondary objective, to find ways to overcome this problem for real world size problems.

To meet these objectives first the data was partitioned using a block and an interleave scheme. This provides classifiers that focus on local and global features. Following partitioning wavelets were used to enrich the feature space for the classification procedure. A subset of the most important classification features was selected using information entropy. This calculation was also used to determine the division point for binary labeling of each range bin. A polynomial time complexity RST method was developed to compute minimal classifiers for each partition. A fusion formula was developed which fused the classifications for all partitions. This was the HRR rough set classifier.

During this research it was found that which wavelet family was used made no statistical difference. It was also determined that an iterated wavelet transform would, in essence, result in a new wavelet tailored to the given problem. This new wavelet produces a 12 percentage point improvement in classifier performance. The rough set classifier produces 93% probability of correct classification and almost 100% probability of declaration. These results are five percentage points better than the most popular method, the quadratic classifier.

Approved: ____

Dr. Janusz A. Starzyk, Professor