Fusing Marginal Reducts for HRR Target Identification

Dale E. Nelson
Sensors Directorate, Air Force Research Laboratory
Wright Patterson Air Force Base, Ohio 45324, USA

Janusz A. Starzyk
Department of Computer and Electrical Engineering, Ohio University
Athens, Ohio 45701, USA

ABSTRACT

Rough set theory has not been applied to automatic target recognition (ATR) problems because the problems of interest were too large. The determination of reducts (classifiers) was a problem whose solution grew in exponential time with the number of range bins. This paper introduces a method which allows the determination of reducts in quadratic time and a method of partitioning the problem (reducing the number of range bins being considered) so that ATR problems can be solved in a reasonable time. A method of fusing the individual classifier results, even though they may not have performed well on the training set (marginal reduct) is introduced. This fusion of marginal reducts yields a synergistic result that produces a well performing classifier.

Keywords: Rough Sets, Automatic Target Recognition, High Range Resolution Radar, Fusion, Reduct.

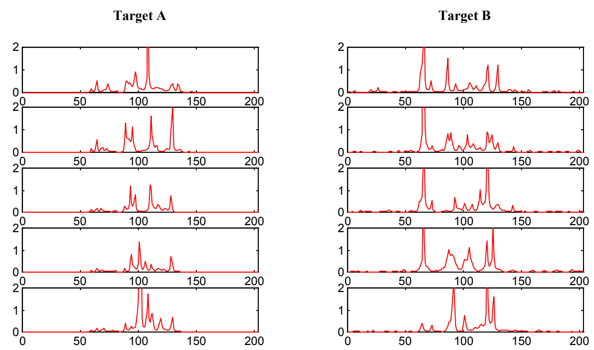
1. PROBLEM

This paper uses High Range Resolution (HRR) radar signals for target classification. A HRR signal is an n-dimensional vector $s = (y_1, y_2, \ldots, y_n)$, and $y_i \in \{0, 1, \ldots, 255\}$. The HRR radar provides a 1-D picture of what the sensor is looking at. HRR signals are particularly hard to use for target recognition, partly because the 3-D world is projected onto just one dimension. When this is done, there are many ambiguities created which must be resolved

A further complication to target identification using HRR is that the signals change considerably with only small changes in azimuth and elevation. This is illustrated in Figure 1 [1].

The signals shown in Figure 1 are from two different targets. The signals shown for each target were taken at 200 msec intervals. The significant variations in a short time span illustrate how difficult it would be to construct a target identification system based on these signals.

Target identification systems are normally specified in terms of goals for performance. Two of the goals used are probability of correct classification (Pcc) and probability of declaration (Pdec). The automatic target recognition (ATR) system may chose to declare or not declare a HRR return as a target. If the ATR system declares on every target, Pcc will be low while Pdec will be high. Conversely if the system only declares when it is absolutely sure, Pcc will be high and Pdec will be low.



The goal for this research effort is a Pcc > 95% and a Pdec > 85%. However we would be willing to sacrifice a small amount of Pcc for a large increases in Pdec.

Figure 1. A comparison of two HRR target signals.

2. 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 ±25° and elevations of -20 ° to 0 ° in one-degree increments as illustrated in Figure 2.

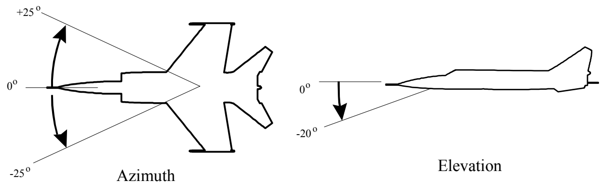


Figure 2. Azimuth and Elevation Ranges

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.

Multiple Data Partitions

One of the problems with rough set classification is that the determination of all the reducts (this is described in a later section) is an NP-hard problem [4]. Even using methods described in [5] the HRR ATR problem is too large. Therefore, it is necessary to only use a small subset of the available range bins. We hypothesized that this 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. We selected the range bins to be used 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. The second selection consisted of two sets, the even numbered range bins, and the odd numbered range bins. For the partitions where the data is in fourths, we selected range bins 1-32, 33-64, 65-96, 97-128. 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 size.

Wavelet Transformation

Wavelets are a well-described method for data analysis and identification [6]. Our previous research [2] revealed that using a wavelet transform's coefficients as pseudo range bins (henceforth also called range bins) yielded features that would improve classifier performance. We also found that it did not matter which wavelet was chosen as the choice did not have a statistically significant different classification performance. Therefore, once the data was divided into multiple partitions and multiple selections, the signal values were normalized. This was accomplished by dividing each signal value by the 2 norm across the signal's range bins. The 2 norm is defined as:

$$N = \left(\sum_i |y_i|^2 \right)^{1/2}$$

After the normalization, each of these data sets was transformed using a multi-level wavelet transform using the Haar wavelet [6]. This resulted in the original signal with 128 range bins becoming a signal with 1024 pseudo range bins. A signal with 64 range bins (signal divided in two partitions) becomes a signal with 448 pseudo range bins. The signals with 32 range bins (signal divided into four partitions) becomes signals with 192 pseudo range bins. Finally, signals with 16 range bins (signal divided into eight partitions) become signals with 80 pseudo range bins.

Binary Entropy Labeling

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. For this effort we chose to use a binary labeling based on multi-class information entropy. Assume a range bin across all training signals is defined as:

$$u_t = \{x_1, \dots, x_n\}$$

Let

$$C_{xt} \equiv \{x_i < x \mid x_i \in u_t\}$$

Rather than forcing an assumed distribution on the data we can use an approximation to the probability density function to obtain the probabilities. Thus we can define the probability at each point x as the quotient of cardinalities:

$$P_{xt} = \frac{|C_{xt}|}{|u_t|}$$

And two more probabilities as:

$$P_{i_x} = \frac{P_{xi} P_{xj}}{\sum_i P_{xi}}$$

$$\overline{P_{i_x}} = \frac{(1 - P_{xi})(1 - P_{xj})}{\sum_i P_{xi}}$$

Where the index i is defined as :

$$i = 6(j-1) + (k-1) + 1$$

Now we define the multi-class entropy for our six class problem as:

$$E \equiv \sum_{j=1}^{6} P_{jx} \log P_{jx} + \sum_{j=1}^{6} \overline{P}_{jx} \log \overline{P}_{jx}$$

$$- \sum_{l=1}^{36} P_{l_x} \log P_{l_x} - \sum_{l=1}^{36} \overline{P}_{l_x} \log \overline{P}_{l_x}$$

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 the target classes. Since the training signals have been labeled, we may find identical signals that are of the same target class (duplicates). We remove these for computational efficiency, as these are redundant. We may also find signals representing different target classes that are identical (ambiguous). We remove these, as they are confusing when constructing a target classifier. What remains are signals which are unique to each target class. These signals are the basis for the decision table used in rough set classification.

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 range bin in the training signal. Using this would provide for a sharp labeling of the data. Values close to this division point could possibly be "mislabeled" due to noise. Therefore we have added a provision in the classifier to provide a buffer zone around the division point. This buffer zone is defined:

$$d = b * \min(x_d - \min(x_i), \max(x_i) - x_d)$$

where d is the distance from the division point, b is the amount of the smallest distance to be used, x_d is the division point, and x_i are the range bin values. 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"; that range bin will not be considered in the classification process for that signal.

Range Bin Selection

If possible, the perfect solution would be to use all of the range bins. However even with the partitioning of the data, the number of range bins was greater than could be handled in a reasonable time. In our research we determined that a practical number of range bins, considering the computer time required, was 50. We used the value of the minimum relative entropy for each range bin to select the 50 best (largest maximum mutual information) range bins to be used for building the classifier. This means that when we use the transformed original signal we will use 50 range bins. When we use two data partitions we will have 100 range bins. When four partitions are used there will be 200 range bins used. Finally when there are eight partitions there will be 400 range bins used. Even in quadratic time, 400 range bins would require too much time if they were used all at one time. With the partitioning method, however, this many range bins can be used.

3. ROUGH SET CLASSIFICATION

It is not the purpose of this paper to be a tutorial of rough set theory. An introduction to rough set theory may be found in [3]. However, some basic concepts need to be introduced in order to understand the contribution of this research. The set of all labeled training signals forms a table consisting of 1s and 0s. Each row corresponds to a given target type. In many cases it is possible to use a subset of the entire signal to distinguish among the different target classes. For example, it may be possible to use range bins 1 through 20 and be able to uniquely classify each signal in the training set. This is called a **reduct**. This term comes from the idea that we have reduced the size of the table without reducing the information contained in it (i.e.; the ability to uniquely classify all the signals). There may be no reducts or there be many reducts. It should be noted that a reduct may not contain another reduct. The range bins common to all reducts are called the **core**. The procedure to determine reducts and the core may be found in [3] and [5].

Multiple Classifiers

One of the major problems in applying rough set theory to classification is that determination of all the reducts has been shown to be NP-hard [4]. Because it is easier to match a few range bins rather than many, we believe that better classification performance is achieved through using **minimal reducts**, that is, reducts that use the fewest range bins. The problem is how to achieve adequate performance of the classifier with the partitioned data. We anticipate that training goals can be easily met, or exceeded, on the training data and yet still not be met on the test set..

Merging of Marginal Reducts

Once all the minimal reducts have been determined each one is tested against the full training set (all ambiguities and duplicates included) and the performance (probability of correct classification, Pcc, and probability of declaration, Pdec) determined. In some cases a reduct based on a data partition which contains mostly noise will have a low Pcc and a high Pdec. This is called a **marginal reduct**. 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.

The rough set classifier will generate a set of classifiers (reducts) for each of the training files. Some of these classifiers will yield a high probability of correct classification (Pcc) while others will yield a low Pcc. 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:

1. If all the Pcc(s) are zero the weight should be zero.
2. If all the Pcc(s) are one the weight should be one
3. 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.
4. 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 far a given target class is given as :

$$W_t = 1 - \frac{Pcc_{\max} + \left[\sum_{i=1}^{n}(1 - Pcc_i)\right](1 - Pcc_{\max})}{\sum_{i=1}^{n} \frac{1}{1 - Pcc_i + \varepsilon}}$$

Where:

Pmax = Maximum Pcc of all reducts "voting" for target class *t*

Pi = Pcc of each "vote" for target class *t*

n = number of "votes" for target class *t*

ε = small number preventing division by zero.

Using some contrived Pcc values Table 1 illustrates the weight generated for a target class based on the "votes" and their Pcc. As can be seen from the table all the desirable features of the function are achieved. The maximum value (1) and the minimum value (0) are achieved. 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 Pcc values.

Pcc(1)	Pcc(2)	Pcc(3)	Pcc(4)	Pcc(5)	Wt
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
0.80	0.10	0.10	0.00	0.00	0.83

Table 1. Results of Weighting Formula

4. CLASSIFICATION RESULTS

The results of classification based on fusing marginal reducts with the weighting formula can be seen in Table 2 and Table 3. The first column indicates the number of data divisions the original signal was divided into. The second column indicates how the data was selected. If the second column is only a number, it indicates the use of an interleave scheme. For example, if there are two data divisions 1 means the odd range bins and 2 means the even ones. If the value in the second column is followed by "st" or "nd" it means that the range bins are from the first block, second block, etc. For example, if the number of data divisions are two, 1st means range bins 1-64 and 2nd means range bins 65-128.

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

Table 2. Fusing Training Partitions.

Table 2 shows a dramatic improvement achieved by fusing the results of the various classifiers as compared to direct classification results (row number 1 in the Table 1). In the case of eight data divisions a perfect score is achieved on both Pcc and Pdec. Intuitively one might expect this. However, remember that both duplicates and ambiguities were removed from the training set to develop the classifier. These results were on the entire training set. When all the partitions are combined together both Pcc and Pdec achieve perfect scores. The only way to truly assess the performance is to test the classifier on the test data set. When comparing eight data divisions to one data division remember that we are comparing target recognition results based on 400 range bins against performance based on 50 range bins.

Div.	Sel.	Pcc	Pdec	Pcc	Pdec	Pcc	Pdec	Pcc	Pdec
1	1	0.1667	0.99979						
2	1	0.79095	0.79726						
2	2	0.79381	0.75705	0.81229	0.94424				
2	1st	0.91704	0.73964						
2	2nd	0.43967	0.7954	0.79694	0.94942	0.88116	0.992		
4	1	0.72369	0.80949						
4	2	0.74261	0.85531						
4	3	0.7574	0.83313						
4	4	0.63639	0.75311	0.82266	0.9971				
4	1st	0	0						
4	2nd	0.88051	0.80846						
4	3rd	0.65241	0.76575						
4	4th	0	0	0.83239	0.94983	0.882	0.999		
8	1	0.63722	0.81882						
8	2	0.58947	0.78317						
8	3	0.59995	0.75808						
8	4	0.51747	0.65858						
8	5	0.49859	0.80618						
8	6	0.50673	0.72367						
8	7	0.51172	0.7073						
8	8	0.53631	0.64511	0.77047	0.99979				
8	1st	0	0						
8	2nd	0	0						
8	3rd	0.66444	0.68138						
8	4th	0.81829	0.51451						
8	5th	0.70485	0.41439						
8	6th	0.37494	0.82546						
8	7th	0	0						
8	8th	0	0	0.75351	0.97388	0.84906	0.999	0.923	0.999

Table 3. Fusing Testing Partitions

Although the test scores are not expected to be as good as for the training set we expect a similar improvement when the results of the classifiers based on the marginal reducts are fused. The results from the test set are shown in Table 3. We see that as the various partitions are combined, the Pcc is not always greater than the greatest Pcc of the individual partitions. This is due to the use of the probabilities from the training set in weighting. In a real world problem there is no way to assess the performance on an unknown signal thus requiring the use of the performance from the training set. Nevertheless we find that the performance increases to an acceptable level as we combine more and more partitions.

We see that the final result of 92% Pcc is below the desired value of 95%. However, Pdec is very close to 100%. We can either accept this result or through minor adjustments of some of the tolerances in the system, we can trade some Pdec for an increase in Pcc. We are willing to accept the small decrease in Pcc for almost 100% Pdec. In the real world this decision would be made under the rules of engagement.

5. SUMMARY

We have shown that a multi-partitioned data set may be used so that rough set theory can be applied to construct a classifier. In order to meet performance goals it is necessary to fuse the results of all the minimal reducts. We have introduced a fusion formula that performs this operation yielding a classifier that has acceptable performance. Test results confirmed our claim of superior performance over direct classification based on fusing all data partitions.

6. REFERENCES

[1] R. Mitchell,, J. Westerkamp, “Robust statistical feature based aircraft identification”, IEEE Transactions on Aerospace and Electronic Systems, Vol 35, No. 3, 1999.

[2] D. Nelson, J. Starzyk, “Iterative Wavelet Transformation and Signal Discrimination for HRR Radar Target Recognition”, IEEE Systems, Man, and Cybernetics, in review.

[3] Z. Pawlak, Rough Sets Theoretical Aspects of Reasoning about Dat*a*, Kluwer Academic Publishers 1991.

[4] A. Skowron, J. Stepaniuk, “Towards an approximation theory of discrete problems: Part I., Fundamenta Informaticae Vol. 15, No. 2, 1991, pp. 331-362

[5] J. Starzyk, D. Nelson, K. Sturtz, “A Mathematical Foundation for Improved Reduct Generation in Information Systems, Knowledge and Information Systems, Vol. 2, No. 2, May 2000.

[6] G. Strang, T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, 1996