

Feature Selection for Target Detection in SAR Images

Bir Bhanu, Yingqiang Lin and Shiqin Wang
Center for Research in Intelligent Systems
University of California, Riverside, CA 92521, USA

Abstract

A genetic algorithm (GA) approach is presented to select a set of features to discriminate the targets from the natural clutter false alarms in SAR images. Four stages of an automatic target detection system are developed: the rough target detection, feature extraction from the potential target regions, GA based feature selection and the final Bayesian classification. Experimental results show that the GA selected a good subset of features that gave similar performance to using all the features.

Key words: Feature selection, genetic algorithm, ATR system, target detection.

1. Introduction

Automatic detection of potential targets in SAR imagery is an important problem. A CFAR (constant false alarm rate) detector is commonly used to “prescreen” the image to localize the possible targets. Generally, targets correspond to bright spots caused by strong radar return from natural or man-made objects. Parts of the imagery that are not selected are rejected from further computation. In the next stage of processing, regions of interest are further examined to distinguish man-made objects from natural clutter. Finally, a classifier such as a Bayesian classifier, a template matcher or a model-based recognizer is used to reject man-made clutter.

The goal of feature selection is to find the subset of features which produces the best detection/recognition accuracy and requires the least computational effort. Feature selection is important to target detection and recognition systems mainly for two reasons:

First, using more features can increase system complexity, yet it may not always lead to higher detection/recognition accuracy. Sometimes, many features are available to a detection/recognition system. However, these features are not independent and may be correlated. A bad feature may greatly degrade the performance of the system. Thus, selecting a subset of good features is important.

Second, using fewer features can reduce the computational cost, which is important for real-time applications. Also it may lead to better classification accuracy due to the finite sample size effect.

Genetic algorithms (GAs) are widely used in image processing, pattern recognition and computer vision. They are used to generate structural elements in mathematical morphology, to select good parameters for object detection and recognition, to generate image filters for target recognition, to select training data for neural networks, etc. GAs are also used to automatically determine the relative importance of many different features and to select a good subset of features available to the system

The focus of this paper is to select a minimal set of features to distinguish targets from natural clutter. The approach is based on a closed loop system involving GA based feature selection and a Bayesian classifier. GA uses an appropriate fitness function that combines the number of features to be used and the error rate of the classifier. The results are presented using real SAR images. The experimental results show that the subset of features selected by GA can greatly reduce the computational cost while at the same time the detection accuracy is maintained.

Section 2 presents the related research and the contribution of this paper. Section 3 describes the approach, the prescreener used to detect potential target regions, the features for target discrimination, feature evaluation criteria and the application of GAs to feature selection. Experimental results are presented in Section 4 and Section 5 provides the conclusion of the paper.

2. Related Research

Chaikla and Qi [1] consider the importance of the domain in eliminating noisy features for feature selection. They present an approach to designing a multi-objective fitness function for the genetic algorithm. The results show that the proposed fitness function can perform more satisfactorily than the traditional one. Emmanouilidis et al. [2] discuss the use of multicriteria genetic algorithms for feature selection. The algorithm is shown to yield a diverse population of alternative feature subsets with various accuracy and complexity trade-off. It is applied to select features for performing classification with fuzzy models and is evaluated on real-world data sets. Estevez et al. [3] propose a genetic algorithm for selecting features for neural network classifiers. Their algorithm is based on a

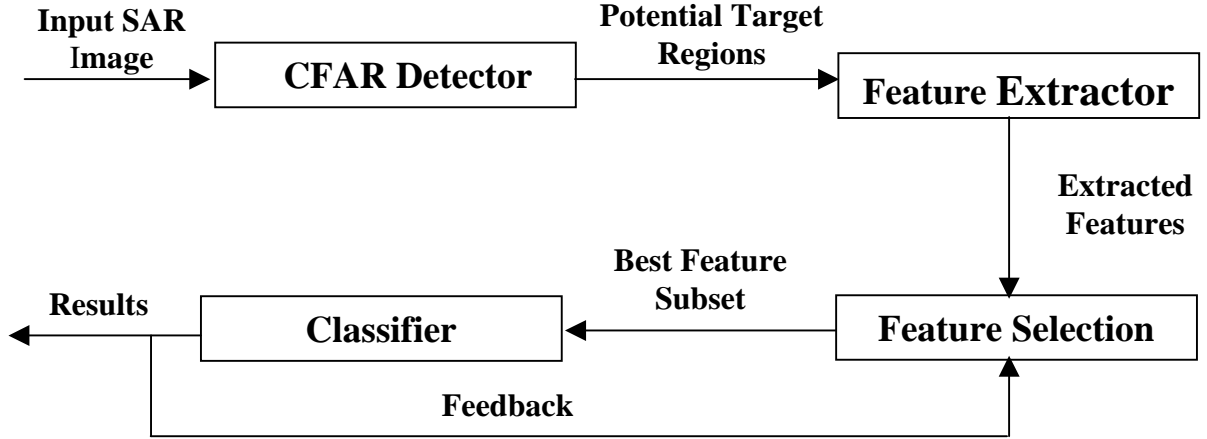


Fig. 1. System diagram for feature selection.

niching method to find and maintain multiple optima. They also introduce a new mutation operator to speed up the convergence of the genetic algorithm. Rhee and Lee [10] present an unsupervised feature selection method using a fuzzy-genetic approach. The method minimizes a feature evaluation index which incorporates a weighted distance used to rank the importance of the individual features. Matsui et al. [6] use genetic algorithm to select the optimal combination of features to improve the performance of tissue classification neural networks and apply their method to problems of brain MRI segmentation to classify gray matter/white matter regions.

In this paper, we use genetic algorithm to select a good subset of features used for target detection in SAR images. The target detection task involves the selection of a subset of features to discriminate SAR images containing targets from those containing clutter. Our method is a novel combination of genetic algorithm based optimization of a criterion function that involves classification error and the number of features that are used for the discrimination of targets from natural clutter in SAR images. The criterion function is optimized in closed-loop with a Bayesian classifier. As compared to this work, the feature selection presented in [5, 8] for target vs. natural clutter discrimination is not optimal but ad hoc.

3. Technical Approach

The purpose of the genetic algorithm (GA) based feature selection approach presented in this paper is to select a set of features to discriminate the targets from the natural clutter false alarms in SAR images. The approach includes four stages: rough target detection, feature extraction from the potential target regions, feature selection based on the training data and the final discrimination. The first two stages are based on the Lincoln Lab ATR system [5, 7, 8]. In the feature selection stage, we use GAs to select a best feature subset, defined as a particular set of features which is the best in discriminating the target from the natural clutter false alarm. The diagram for feature selection is given in Fig. 1.

3.1. CFAR Detector

A two-parameter CFAR detector is used as a prescreeener to identify potential targets in the image on the basis of radar amplitude. A guard area around a potential target pixel is used for the estimation of clutter statistics. The amplitude of the test pixel is compared with the mean and standard deviation of the clutter according to the following rule:

$$\frac{X_t - \hat{\mu}_c}{\hat{\sigma}_c} > K_{CFAR} \Rightarrow \text{target}, \text{ otherwise clutter} \quad (1)$$

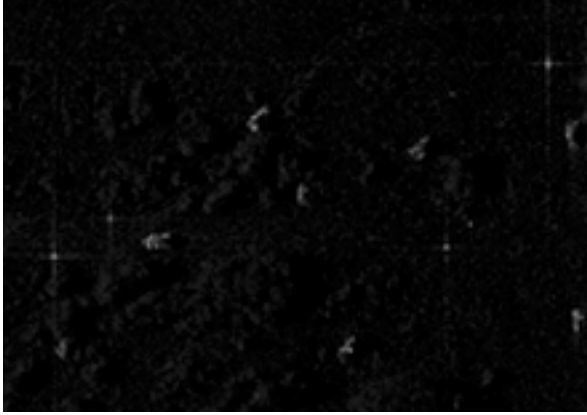
where X_t is the amplitude of the test pixel, $\hat{\mu}_c$ is the estimated mean of the clutter amplitude, $\hat{\sigma}_c$ is the estimated standard deviation of the clutter amplitude, and K_{CFAR} is a constant threshold value that defines the false-alarm rate.

Only those test pixels whose amplitude is much higher than that of the surrounding pixels are declared to be targets. The higher we set the threshold value of K_{CFAR} , the more a test pixel must stand out from its background for it to be declared as a target. Because a single target can produce multiple CFAR detections, the detected pixels are grouped together if they are within a target-sized neighborhood. The CFAR detection threshold in the prescreeener is set relatively low to obtain a high initial probability of detection for the target data. It is the responsibility of the discriminator to capture and reject those escaping clutter false alarms from the prescreen stage. An example SAR image and corresponding detection results are shown in Fig. 2.

3.2. Feature Extractor

First, we use a target-size rectangular template to determine the position and orientation of the detected target [4]. The algorithm slides and rotates the template until the energy within the template is maximized. Then we extract a set of features from the target-sized

template or the region of interest. By using this set of features [2], we try to discriminate the targets from the natural clutter.



(a) Example SAR image.



(b) Detection result.

Fig. 2. SAR image and CFAR detection result.

3.2.1. The standard-deviation feature

The standard deviation of the data within the template is a statistical measurement of the fluctuation of the pixel intensities. If we use $P(r, a)$ to represent the radar intensity in power from range r and azimuth a , the standard deviation can be calculated as follows:

$$\sigma = \sqrt{\frac{S_2 - \frac{S_1^2}{N}}{N - 1}} \quad (2)$$

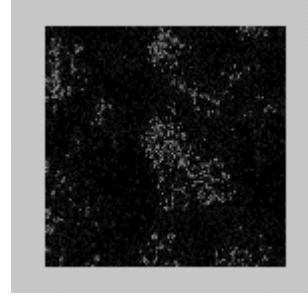
where

$$S_1 = \sum_{r, a \in \text{region}} 10 \log_{10} P(r, a)$$

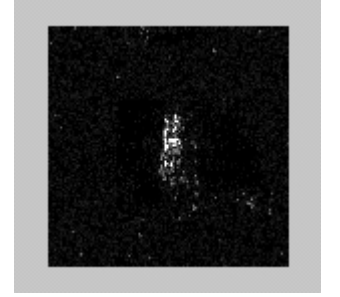
$$S_2 = \sum_{r, a \in \text{region}} [10 \log_{10} P(r, a)]^2$$

and N is the number of points in the region.

Targets usually exhibit much larger standard deviation than the natural clutter, as illustrated by Fig. 3.



(a) A typical natural clutter image with standard deviation 4.5187.



(b) A typical target image with standard deviation 5.2832.

Fig. 3. Example of the standard deviation feature.

3.2.2. The fractal dimension feature

The fractal dimension of the pixels in the region of interest provides information about the spatial distribution of the brightest scatterers of the detected object. It complements the standard-deviation feature, which depends only on the intensities of the scatterers, not on their spatial locations.

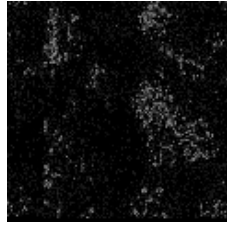
The first step in applying the fractal-dimension concept to a radar image is to select an appropriately sized region of interest, and then convert the pixel values in the region of interest to binary. One method of performing this conversion is to select the N brightest pixels in the region of interest and convert their values to 1, while converting the rest of pixel values to 0. Based on these N brightest pixels, we approximate the fractal dimension by using the following formula:

$$\text{dim} = -\frac{\log M_1 - \log M_2}{\log 1 - \log 2} = \frac{\log M_1 - \log M_2}{\log 2} \quad (3)$$

where M_1 represents the minimum number of 1-pixel-by-1-pixel boxes that cover all N brightest pixels in the region of interest (This number is obviously equal to N) and M_2 represents the minimum number of 2-pixel-by-2-pixel boxes required to cover all N brightest pixels.

The bright pixels for a natural clutter tend to be widely separated, thus produce a low value for the fractal dimension, while the bright pixels for the target tend to be closely bunched, thus we expect a high value for the fractal dimension, which is illustrated by Fig. 4. Fig. 4. (a) shows a natural clutter image chip. In Fig. 4. (b), the 50 brightest pixels from this natural clutter are relatively isolated, and 46 2x2-pixel boxes are needed to cover them, which results in a low fractal dimension of 0.29. Fig. 4. (c) shows a target image chip. In Fig. 4. (d), the 50 brightest pixels from the target image are tightly clustered, and 22 2x2-pixel boxes are needed to

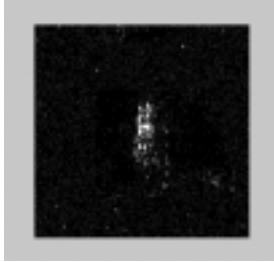
cover them, which results in a high fractal dimension of 1.2.



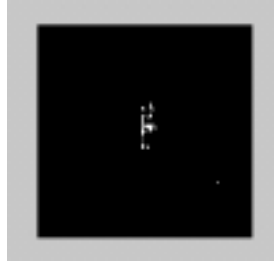
(a) Natural clutter image.



(b) 50 brightest pixels in (a).



(c) Target image.



(d) 50 brightest pixels in (c).

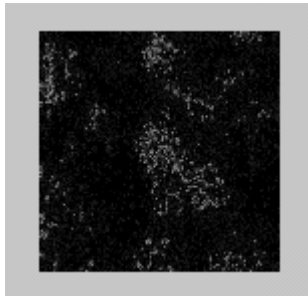
Fig. 4. Example of the fractal dimension feature.

3.2.3. Weighted-Rank Fill Ratio Feature

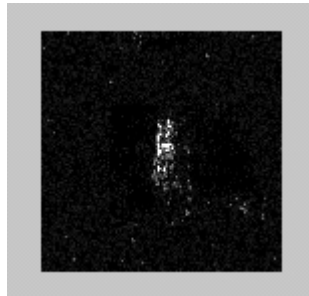
This textual feature measures the percentage of the total energy contained in the brightest scatterers of a detected object. We define the weighted-rank fill ratio as follows:

$$\eta = \frac{\sum_{k \text{ brightest pixels}} P(r, a)}{\sum_{\text{all pixels}} P(r, a)} \quad (4)$$

This feature attempts to exploit the fact that power returns from most targets tend to be concentrated in a few bright scatterers, whereas power returns from natural-clutter false alarm tend to be more diffuse. Fig. 5 shows the fill ratio we get from typical examples of target and clutter images.

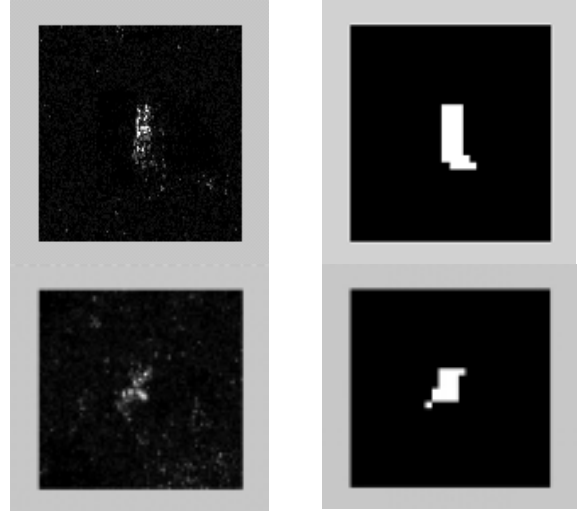


(a) A typical natural clutter image and the fill ratio is 0.2321.

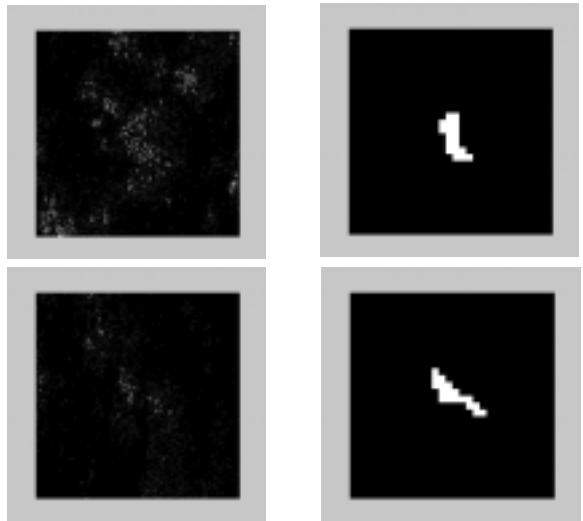


(b) A typical target image and the fill ratio is 0.3861.

Fig. 5. An illustration of the fill ratio feature.



(a) The left-hand side figures represent the target images and right-hand figures represent their corresponding morphological blobs.



(b) The left-hand side figures represent the clutter images and right-hand figures represent their corresponding morphological blobs.

Fig. 6. Examples of the size feature for (a) targets and (b) clutter.

3.2.4. Size-related feature

The three size-related features utilize only the binary image created by the morphological operations.

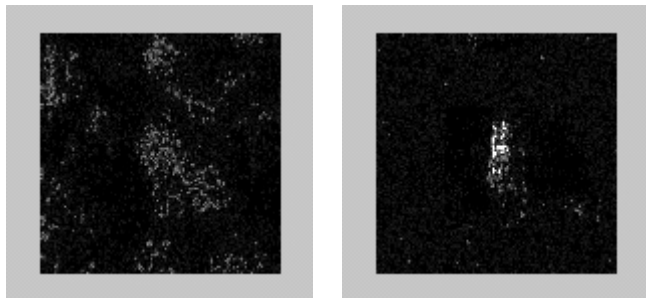
1. The mass feature is computed by counting the number of pixels in the morphological blob.
2. The diameter is the length of the diagonal of the smallest rectangle that encloses the blob.
3. The square-normalized rotational inertia is the second mechanical moment of the blob around its center of mass, normalized by the inertia of an equal mass square.

In our experiments, we found the size features are not effective in scenarios where the targets are partially occluded or hidden. After the prescreeener stage, the size and the shape of the detected morphological blob can be arbitrary. For the clutter, there is also no ground to assert that the resulting morphological blob will exhibit a certain amount of coherence. The experimental results in Fig. 6 show the arbitrariness of the morphological blobs for the targets as well as the clutter.

3.2.5. The contrast-based features

The CFAR statistic is computed for each pixel in the target-shaped blob to create a CFAR image. Then the three features can be derived as follows:

1. The maximum CFAR feature is the maximum value in the CFAR image contained with the target-sized blob.
2. The mean CFAR feature is the average of the CFAR image taken over the target-shaped blob.
3. The percent bright CFAR feature is the percentage of pixels within the target-sized blob that exceed a certain CFAR value.



(a) A typical natural clutter image and the three CFAR features are 10.32, 2.37 and 0.0042, respectively. (b) A typical target image and the three CFAR features are 55.69, 5.53 and 0.15, respectively.

Fig. 7. Examples of the CFAR features.

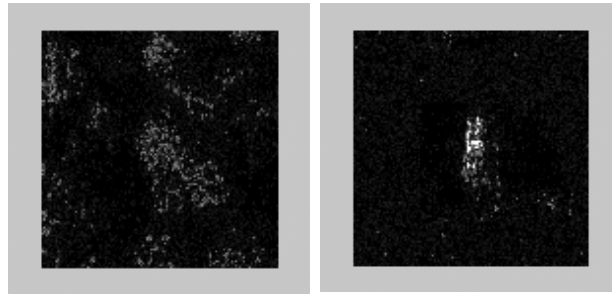
We can see from Fig. 7 that the CFAR feature values for the target are much larger than those for the natural clutter false alarm.

3.2.6. The count feature

The count feature is very simple; it counts the number of pixels that exceeded the threshold T and normalize this value by the total possible number of pixels in a target blob. The threshold T is set to the quantity corresponding to the 98th percentile of the surrounding clutter. The Fig. 8 shows the count feature values obtained from a pair of typical target and clutter images.

We can see from Fig. 8 that the count feature value for the target is much larger than that for the nature clutter false alarm. This makes sense because the intensity values of the pixels belonging to the target stand out from the surrounding

clutter, while the natural clutter false alarms do not have this property.



(a) A typical natural clutter image and the count feature value is 0.1375. (b) A typical target image and the count feature value is 0.6.

Fig. 8. The illustration of the count feature.

3.3. Feature Evaluation and selection

Adding more features does not necessarily improve discrimination performance. An important goal is to choose the best set of features from the discrimination features given in section 3.2 above. Before we do the feature selection, it is appropriate to give a set of feature evaluation criteria, which measure the discrimination capability of each feature or the combination of several features.

- **Divergence:** Divergence is basically a form of the Kulback-Liebler distance measure between density functions. If we assume that the target as well as the natural clutter feature vectors follow the Gaussian distributions respectively, that is, $N(\mathbf{u}_1, \Sigma_1)$ and $N(\mathbf{u}_2, \Sigma_2)$, the divergence can be computed as follows:

$$d_{12} = \frac{1}{2} \text{trace} \left\{ \Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2I \right\} + \frac{1}{2} (\mathbf{u}_1 - \mathbf{u}_2)^T (\Sigma_1^{-1} + \Sigma_2^{-1}) (\mathbf{u}_1 - \mathbf{u}_2) \quad (5)$$

- **Scatter Matrices:** One major drawback of the divergence d_{12} is that it is not easily computed, unless the Gaussian assumption is employed. For SAR imagery, the Gaussian assumption itself is in question. So we consider a simpler criterion that is based upon the information related to the way feature vector samples are scattered in the l -dimensional feature space. At the beginning, we define two kinds of scatter matrices, that is, within-class scatter matrix and between-class scatter matrix. Within-class scatter

matrix $S_w = \sum_{i=1}^M P_i S_i$, where S_i is the covariance

matrix for class ω_i and P_i is the a priori probability of class ω_i . S_ω matrix measures how feature vector samples are scattered within each class.

Between-class scatter matrix is defined as follows: $S_b = \sum_{i=1}^M P_i (\mathbf{u}_i - \mathbf{u}_0)(\mathbf{u}_i - \mathbf{u}_0)^T$, where \mathbf{u}_0 is the

global mean vector and \mathbf{u}_i is the mean for each class, $i = 1, \dots, M$. The between-class scatter matrix measures how the feature vector samples are scattered between different classes.

Based on the different combinations of these two scatter matrices, a set of class separability criteria can be derived. We choose to use one of them that can be defined as

follows: $J = \frac{|S_b|}{|S_w|}$. If the feature vector samples within each

class are scattered compactly and different classes are far away from one other, we expect that J value would be high. This also implies that the features we choose have large discrimination capability.

- **Feature vector evaluation using a classifier:** Another method for feature evaluation depends on the specific classifier. The task of feature selection is to select or determine a set of features, that when fed into the classifier, will let the classifier achieve the best performance. So it makes sense to relate the feature selection procedure to the particular classifier used. During the training time, we have all the features extracted from the training data. What we can do is to select a subset of these features and feed them into the classifier and see the classification result. Then the goodness of each feature subset is indicated by their classification error rate.

3.3.1 GAs for Feature selection

The genetic algorithm is an optimization procedure that operates in binary search spaces (the search space consists of binary strings). A point in the search space is represented by a finite sequence of 0's and 1's, called a *chromosome*. The algorithm manipulates a finite set of chromosomes, the *population*, in a manner resembling the mechanism of natural evolution. Each chromosome is evaluated to determine its "fitness," which determines how likely the chromosome is to survive and breed into the next generation. The probability of survival is proportional to the chromosome's fitness value. Those chromosomes which have higher fitness values are given more chances to "reproduce" by the processes of *crossover* and *mutation*. The function of crossover is to mate two parental chromosomes to produce a pair of offspring chromosomes. In particular, if a chromosome is represented by a binary string, crossover can be implemented by randomly choosing a point, called the crossover point, at which two chromosomes exchange their parts to create two new chromosomes. Mutation randomly perturbs the bits of a single

parent to create a child. This procedure can increase the variability of the population. A mutation can be created by flipping at random one or more bits in the chromosome.

In this work, there are 10 features as described earlier. Each feature is represented as a bit in the genetic algorithm. There are 1024 possible combination of these features.

- **Applying GA for feature selection:** We use GA to seek the smallest (or the least costly) subset of features for which the classifier's performance does not deteriorate below a certain specified level [9, 11]. The basic system framework is shown in the Fig. 1.

When the error of a classifier is used to measure the performance, a subset of features is defined as feasible if the classifier's error rate is below the so-called *feasibility threshold*. We search for the smallest subset of features among all feasible subsets. During the search, each subset can be coded as a d -element bit string (d is the initial number of features). The i th element of the bit string assumes 0 if the i th feature is excluded from the subset and 1 if it is present in the subset. We use the following *penalty function* [11]:

$$p(e) = \frac{\exp((e-t)/m) - 1}{\exp(1) - 1} \quad (6)$$

where e is the error rate, t is the feasibility threshold and m is called the "tolerance margin". This penalty function can be seen Fig 9, where $t = 0.1$ and $m = 0.05$.

We can see from Fig. 9, if $e < t$, $p(e)$ is negative and as e approaches zero, $p(e)$ slowly approaches its minimal value. Note also that $p(t) = 0$ and $p(t+m) = 1$. For greater values of the error rate, this penalty function quickly rises toward infinity. We modify the penalty function in equation (6) by adding the number of features in the evaluated subset to produce the score J :

$$J = \gamma \times \text{number_of_features} + (1-\gamma) \times p(e) \quad (7)$$

where γ is the weight between [0, 1] that we give to the number of features.

The last thing we need to do is to design a fitness function used to evaluate the fitness of each chromosome c_i . Since in our case, we look for the minimum of the score, we define the fitness function as

$$f(c_i) = -J(c_i) \quad (8)$$

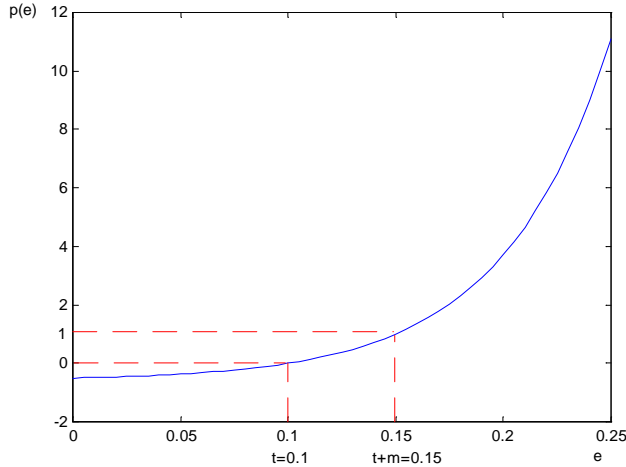


Fig. 9. The penalty function.

4. Experimental Results

All the SAR images used in these experiments are downloaded from a website of the MIT Lincoln Lab <http://www.mbvlab.wpafb.af.mil/public/sdms/menu/sitemapframe.htm>. From these SAR images, 40 target chips (containing target) and 40 clutter chips (containing clutter) are generated. Some of the target and clutter chips are used as training data in our experiments and the rest are used as testing data. The GA selects a good subset of features from the 10 features described previously to classify an image chip into either a

target or clutter. Based on the data we have, we use the CFAR detector in the prescreeener stage to detect the potential target regions. Since we know the ground truth, we know which one is the real target and which one is the clutter false alarm among the potential target regions detected. This allows us to construct a set of training data (training target data and training natural clutter false alarm data) for the feature selection. Then we extract a set of 10 features from each potential target region and do the feature selection. Finally in the testing stage we use the selected features to discriminate the targets from the natural clutter false alarm

Fig. 10 shows the feature extraction interface. On the left-hand side of the interface, we can choose any combination of the features and the number of the training samples. The plot in the middle of right-hand side visualizes the target samples as well as the natural clutter false alarm samples in the feature space. At the bottom, we give a set of feature evaluation values by using the divergence, scatter matrix and the Bayesian Classifier. We can also use the interface to browse through the target training images and the natural clutter raining images. If we are interested in a particular training image, by clicking on that image, we can get the values of the features we extract from that image.

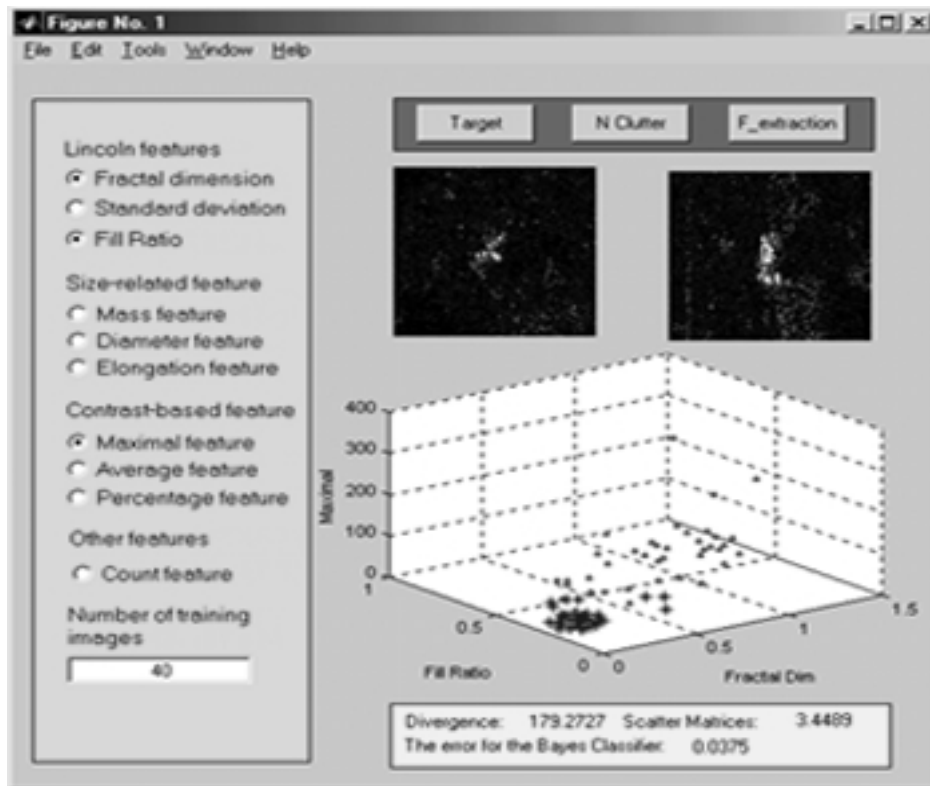


Fig. 10. The feature extraction interface.

Table 1. GA feature selection training and testing results.

Training Data		Feature selected	Number of features	Error rate	Testing Data		Error rate	Number of errors	
Target number	Clutter number				Target number	Clutter number		Target error	Clutter error
15	15	1101001001	5	0	25	25	0.1	1	4
20	20	1011101001	6	0	20	20	0.075	0	3
25	25	1011101001	6	0	15	15	0.1	1	2
30	30	0001001111	5	0	10	10	0.1	1	1

	Target	Clutter
Target	24	1
Clutter	4	21

(a) 15 training target and clutter chips.

	Target	Clutter
Target	20	0
Clutter	3	17

(b) 20 training target and clutter chips.

	Target	Clutter
Target	14	1
Clutter	2	13

(c) 25 training target and clutter chips.

	Target	Clutter
Target	9	1
Clutter	1	9

(d) 30 training target and clutter chips.

Fig. 11. Confusion matrices of GA for various training and testing experiments.

	Target	Clutter
Target	23	2
Clutter	2	23

(a) 15 training target and clutter chips.

	Target	Clutter
Target	20	0
Clutter	5	15

(b) 20 training target and clutter chips.

	Target	Clutter
Target	14	1
Clutter	3	12

(c) 25 training target and clutter chips.

	Target	Clutter
Target	8	2
Clutter	0	10

(d) 30 training target and clutter chips.

Fig. 12. Confusion matrices for various training and testing experiments using all 10 features.

The feature extraction interface is used to conduct a set of experiments to evaluate the discrimination power of the features. If the features are uncorrelated, we can select the l best features individually and form a l -dimensional feature vector features. This is called scalar feature selection. But in reality, there are always some correlations between different

features. So we consider the classification accuracy of feature vectors.

For our GA feature selection framework, we adopt a Bayesian Classifier to classify the training data and the resulting error rate is used as the feedback into the

feature selection algorithm. We fix the parameters of the GA algorithm: crossover rate 0.6, mutation rate 0.1 and the size of the population 100. γ is set to 0.5 in equation (7).

A series of four experiments was conducted to test the effectiveness of the GA feature selection. In the first experiment, 15 target and 15 clutter chips are used in training and the rest are used in testing; in the second experiment, 20 target and 20 clutter chips are used in training and the rest are used in testing; in the third experiment, 25 target and 25 clutter chips are used in training and the rest are used in testing; in the fourth experiment, 30 target and 30 clutter chips are used in training and the rest are used in testing. The experimental results are reported in Table 1 and confusion matrices are shown in Fig. 11.

In order to further evaluate the features selected by GA, another four experiments were performed, where all the 10 features are used to distinguish the target chips from the clutter chips. Similar to the experiments reported above, the number of training target chips and clutter chips are 15, 20, 25 and 30, respectively. Fig. 12 shows the confusion matrices that resulted from these four experiments.

From Fig. 11 and Fig. 12, it is easy to see that the performance of the features selected by GA is similar to that of using all the 10 features. So GA is capable of selecting a good subset of features from all the features available. However, by using fewer features, a lot of computation is saved.

5. Conclusions

In this paper, we introduced the GA feature selection algorithm into a specific application domain to discriminate the targets from the natural clutter false alarm in SAR images. Rough target detection, feature extraction, GA feature selection and final discrimination are successfully implemented and good results are obtained. Our experimental results show that the GA selected a good subset of features that gave similar performance to using all the features. In the future, we plan to extend this approach to additional features and more complex background clutter.

Acknowledgment: This work was supported in part by grant F33615-99-C-1440. The contents and information do not necessarily reflect the position or policy of U.S. government.

References

[1] N. Chaikla and Y. Qi, "Feature Selection Using the Domain Relationship with Genetic Algorithms," *Knowledge and Information Systems*, vol.1, (no. 3), Springer-Verlag, pp. 377-390, Aug. 1999.

[2] C. Emmanouilidis, A. Hunter, J. MacIntyre, and C. Cox, "Multiple-Criteria Genetic Algorithms for Feature Selection in Neuro-Fuzzy Modeling," *Proc. Int. Joint Conf. on Neural Networks*, vol. 6, pp. 4387-4392, Piscataway, NJ, USA, 1999.

[3] P.A. Estevez and R.E. Caballero, "A Niching Genetic Algorithm for Selecting Features for Neural Classifiers," *Proc. 8th Int. Conf. on Artificial Neural Networks*, vol. 1, pp. 311-316, Springer-Verlag London, 1998.

[4] S.D. Halversen, "Calculating the Orientation of a Rectangular Target in SAR imagery," *Proc. IEEE National Aerospace and Electronics Conf.*, pp. 260-264, May 1992.

[5] D.E. Kreithen, S.D. Halversen, and G.J. Owirka, "Discriminating Targets from Clutter," *Lincoln Laboratory Journal*, vol. 6, no. 1, pp. 25 - 52, Spring 1993.

[6] K. Matsui, Y. Suganami, and Y. Kosugi, "Feature Selection by Genetic Algorithm for MRI Segmentation," *Systems and Computers in Japan*, vol. 30, (no. 7), pp. 69-78, Scripta technical, June 30 1999.

[7] L.M. Novak, M.C. Burl, and W.W. Irving, "Optimal Polarimetric Processing for Enhanced Target Detection," *IEEE Trans. Aerosp. Electron. Syst.* 29, pp. 234 - 244, 1993.

[8] L.M. Novak, G.J. Owirka, and C.M. Netishen, "Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System," *Lincoln Laboratory Journal*, vol. 6, no. 1, pp. 11 - 24, Spring 1993.

[9] W.F. Punch and E.D. Goodman, "Further Research on Feature Selection and Classification Using Genetic Algorithms," *Proc. 5th Int. Conf. on Genetic Algorithms*, pp. 557-564, 1993.

[10] F.C. Rhee and Y.J. Lee, "Unsupervised Feature Selection Using a Fuzzy-Genetic Algorithm," *Proc. IEEE Int. Fuzzy Systems Conf.*, vol. 3, pp. 1266-1269, Piscataway, NJ, 1999.

[11] W. Siedlecki and J. Sklansky, "A Note on Genetic Algorithms for Large-Scale Feature Selection," *Pattern Recognition Letters*, vol. 10, pp. 335-347, November 1989.