

# Binary Decision Tree Using K-Means and Genetic Algorithm for Recognizing Defect Patterns of Cold Mill Strip

Kyoung Min Kim<sup>1,4</sup>, Joong Jo Park<sup>2</sup>, Myung Hyun Song<sup>3</sup>, In Cheol Kim<sup>1</sup>,  
and Ching Y. Suen<sup>1</sup>

<sup>1</sup> Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1455 de Maisonneuve Blvd. West, Suite GM606, Montreal, Canada H3G 1M8  
{kkm, kiminc, suen}@cenparmi.concordia.ca

<sup>2</sup> Department of Control and Instrumentation Engineering, Gyeongsang National University, 900, Gazwa-dong, Chinju, Gyeongnam, 660-701, Korea

<sup>3</sup> Department of Electric Control Engineering, Sunchon National University, 315, Maegok-dong, Suncheon, Chonnam, 540-742, Korea

<sup>4</sup> Department of Electrical Engineering, Yosu National University, San 96-1, Dunduk-dong, Yeosu, Chonnam, 550-749, Korea

**Abstract.** This paper proposes a method to recognize the various defect patterns of a cold mill strip using a binary decision. In classifying complex patterns with high similarity like these defect patterns, the selection of an optimal feature set and an appropriate recognizer is a pre-requisite to a high recognition rate. In this paper GA and K-means algorithm were used to select a subset of the suitable features at each node in the binary decision tree. The feature subset with maximum fitness is chosen and the patterns are divided into two classes using a linear decision function. This process is repeated at each node until all the patterns are classified into individual classes. In this way, the classifier using the binary decision tree can be constructed automatically, and the final recognizer is implemented by a neural network trained by standard patterns at each node. Experimental results are given to demonstrate the usefulness of the proposed scheme.

## 1 Introduction

To produce a cold mill strip of high quality, it is important to extract the defects on the surface of cold mill strip rapidly in the manufacturing process. So, efficient methods for the recognition and extraction of defect patterns of cold mill strips have been studied[1]. Recently, a pattern recognition method to substitute a defect extraction system using a one dimensional reflected laser signal.

The conventional method to recognize the defect patterns is to extract good features experimentally after preprocessing the image acquired from a CCD camera and then recognizes the patterns in a single step by inputting all the features to a neural network. But this method has two problems when the characteristics of the defect

patterns are considered. Firstly, because the shapes of the defect patterns are complex and irregular, the recognition rate of defect patterns is sensitive to the kinds of selected features. And also despite the good separability of the features, they may interfere with each other when used together. So, the fitness of the selected feature subset cannot be guaranteed if the features are selected experimentally. Secondly, because there exist some similar classes of defect patterns, which can be classified into the same group, classifying all the patterns in only a single step results in a high classification error.

To overcome these problems, we propose a multi-stage classifier like a decision tree, which repeats decisions so as to classify patterns individually. The decision tree classifier makes fast and exact decisions by dividing the complex and global decisions into several simple and local decisions [5][8].

For an efficient and accurate classification, an optimal or near-optimal feature subset within the feature space needs to be selected at each decision node [2]. There are three potential advantages in applying a method of selecting a subset of features for an input to the classification. Firstly, the performance of the classifier can be improved by reducing the possible inputs to a set of relevant uncorrelated variables. Secondly, the selected smaller set of features reduces both the time complexity and the processing time needed to produce the feature set, thus speeding up the response time of the system. Finally, since there is a direct relationship between the dimensionality of a problem and the size of the example set needed to adequately cover the problem space, the reduction of the feature set indicates that a smaller off-line training set can be used with all the secure benefits in terms of data collection and training times.

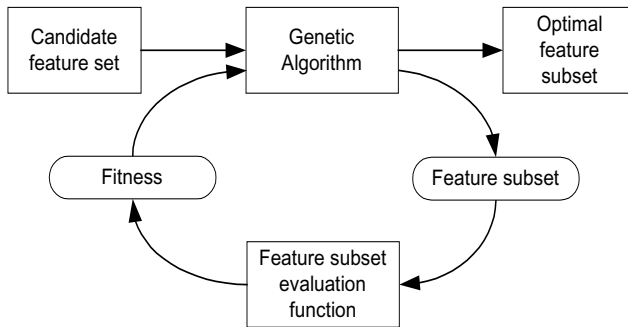
In this paper, GA and K-means algorithm are used to find a subset that yields the lowest error rate of a classifier. This search method has the advantage of producing a nearly optimal solution quickly. The fitness function used to evaluate the selected subsets of features is the error estimator for the linear decision function, which is also produced by GA and K-means algorithm. They make a linear decision function, whose dimension is that of the selected feature space, and searches for a linear decision function, which minimizes the classification error with which the fitness of the selected feature subset is calculated. Finally, the feature subset with maximum fitness is chosen, and the patterns are classified into two classes by the linear decision function. This process is repeated at each node until all patterns are classified respectively into the individual classes. In this way, the binary decision tree classifier is constructed automatically. After constructing the binary decision tree, the final recognizer is accomplished by a neural network, which learns from a set of standard patterns at each node.

This paper introduces the binary decision tree and presents methods of both generating a linear decision function and selecting a feature subset using GA and K-means algorithm. Then, an automatic method of constructing the binary decision tree is described. And finally, the two classifiers are applied to recognize the defect patterns of a cold mill strip.

## 2 Construction of Binary Decision Tree Using GA and K-Means Algorithm

If only the necessary features are used at each node of binary decision tree classifier, both the accuracy and the reliability of the classifier are increased. So the problem is to select a valid feature subset from the entire feature set, the feature selection problem.

GA has a higher probability of finding the global optimized solution than other conventional optimization algorithms because it searches for the multiple global solutions simultaneously. Thus the optimal feature subset can be selected effectively[1][3][4].



**Fig. 1.** Processing block diagram of feature selection

Fig. 1 shows the process of selecting the optimal feature subset by GA. In GA, the chromosomes represent the feature subsets. The fitness of each chromosome is calculated by evaluating the validity of its feature subset, and then the survival probability is determined. According to this probability, the operations of evolution are executed. In this way, new evolved feature subsets are generated. The optimal feature subset is produced by iterating the evolving process. The process of evaluating a feature subset is the most important process in achieving the optimal solution. In this paper, a feature subset is evaluated by the classification error when classifying patterns with the linear decision function that is also generated by GA and K-means algorithm

### 2.1 Evaluation of Feature Subset Using K-Means Algorithm

To solve the problems of one-stage classifier, the classifier that decide the class of the input pattern by repeating two or more decisions successively, is designed and it is called a multi-stage classifier or decision tree classifier.

It is called 'binary decision tree classifier' that has two child nodes at each node. The binary decision tree classifier divides the patterns into two classes with a suitable

feature subset at each node, and this process is iterated until only one pattern class exists in each leaf node.

To select the optimized feature subset, the separability of all the combinations of the features should be evaluated. However, when  $m$  components are selected among  $n$  components, the number of combinations expressed as  $nCm$  becomes a large value even if  $n$  and  $m$  are not large.

There are some searching algorithms which avoid the exhaustive searching like above, which are top-down, bottom-up, branch and bound, and so on. The feature selection problem can be regarded as an optimization problem. So in this paper, firstly, feature selection is executed using K-means algorithm.

It is based on the minimization of a performance index that is defined as the sum of the squared distances from all points in a cluster domain to the cluster center.

## 2.2 Evaluation of Feature Subset and Determination of the Linear Decision Function Using GA

The following method is used to minimize the classification error using GA:

Suppose that the given data set is  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  ( $\mathbf{x}_k \in R^n$  is the number of features), and  $l(j)$  and  $r(j)$  are defined as the minimum and maximum values of the  $j$ -th feature.

$$\begin{aligned} l(j) &= \min_i x_{ij} \\ r(j) &= \max_i x_{ij} \end{aligned} \quad (1)$$

In the case of 2-dimensional space,  $j$  can have the value of 1 or 2 and on the basis of  $l(j)$  and  $r(j)$ , a rectangle can be constructed that can include all data. Inside the rectangle, two points can be selected arbitrarily, connected by a line. From the coefficients of the line function, a 2-dimensional decision function can be obtained as follows.

$$d(\mathbf{x}) = w_1x_1 + w_2x_2 + w_3 = 0 \quad (2)$$

When expanding the 2-dimensional case to the  $n$ -dimensional case, a hyperplane can be formed by selecting  $n$  points in the hyperspace. And values of  $w_1, w_2, \dots, w_{n+1}$  can be found which appear in the linear decision function of the  $n$ -dimensional space, denoted by Eq. (3).

$$\begin{aligned} d(\mathbf{x}) &= w_1x_1 + w_2x_2 + \dots + w_nx_n + w_{n+1} \\ &= \mathbf{w}_0^T \mathbf{x} + w_{n+1} \end{aligned} \quad (3)$$

When matching this concept with a binary string of the GA,  $n$  segments of a binary string indicate one point in the  $n$ -dimensional space. In the  $n$ -dimensional case,  $n$  points should be selected in such a way that a string is composed of  $n^2$  segments.

Supposing that the length of each segment is  $m$ -bit, then the total length of the binary string becomes  $n^2m$ -bit.

GA determines the decision function that minimizes classification error in a given feature space. Since the minimized error varies with the combination of features, the fitness function is constructed to give high fitness for a combination with a small classification error and low fitness for a combination with a large classification error.

### 2.3 Construction of the Binary Decision Tree and the Final Recognizer

Using the method described above, a certain feature subset minimizing classification error is chosen. And patterns are classified into two groups at each node with this feature subset. The binary decision tree is constructed by iterating this process until all the classes of patterns appear independently at each leaf node. Because the binary decision tree is constructed for multiple classes rather than just for two it is better to maintain uniform distribution for two separated groups at each node, which means it is better that two separated groups have similar numbers of classes without partiality.

To quantify this, a balance coefficient is defined using the mean and deviation of classes of a new group, as Eq. (4). If the number of patterns of the two separated groups is similar, the balance coefficients are smaller. In this case, because the depth of the binary tree becomes small, the matching time required for recognizing a pattern decreases. The smallest value of the balance coefficient is  $0$  and the largest value is  $\sqrt{2}$  for the binary tree case.

$$balance = \sqrt{\frac{\sum_{j=1}^h (N_j - \frac{N}{h})^2}{(\frac{N}{h})^2}} \tag{4}$$

In Eq. (4),  $h$  is the number of nodes,  $N$  is the number of input patterns, and  $N_j$  is the number of the patterns included the  $j$ -th node. In this paper, a binary tree is constructed, so  $h$  becomes  $2$ . The fitness function that includes the balance coefficient is defined as.

$$fitness = \frac{1}{1 + w_e \cdot error + w_b \cdot balance} \tag{5}$$

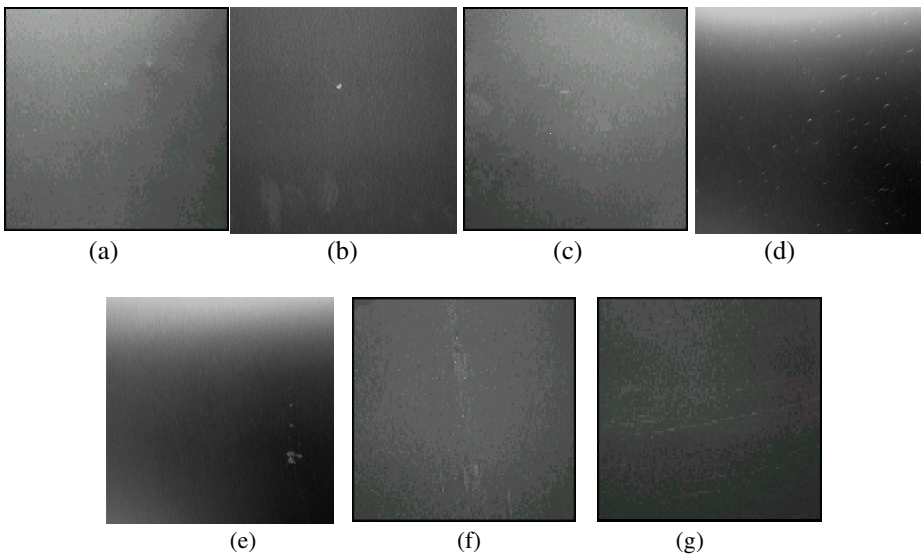
In Eq. (5), *error* and *balance* are the classification error and the balance coefficient between groups, respectively, and  $w_e$  and  $w_b$  are the weights for weighting each parameter. If both the classification error and the balance coefficient have the value  $0$ , fitness has the largest value  $1$ . And the result of the constructed tree can be varied by adjusting of the weights  $w_e$  and  $w_b$ . For example, if a large value is assigned to  $w_b$ , the probability that a more balanced tree structure can be obtained becomes high, while the error rate also becomes high.

After the construction of the binary decision tree, by training BP neural network with the feature subset selected optimally at each node, the final binary tree structured recognizer is realized.

### 3 Classification of the Defects of Cold Mill Strip Using Binary Tree Classifier

#### 3.1 Extraction of the Features of the Defect Pattern

The defect patterns of cold mill strips can be classified into seven classes: Dull, Oil-drop, Slip, Dent, Scale, Dirt, and Scratch. After preprocessing for the acquired image, we extract six candidate features[1] .



**Fig. 2.** Defect patterns of cold mill strip (a) dull (b) oil drop (c) slip (d) dent (e) dirt (f) scale (g) scratch

In this paper, geometrical features are selected as candidate features. They are area, area ratio, and compactness. They are not related to the size and direction of the patterns.

1. *def\_area* : the area of a pattern  
(the number of pixels of a defect pattern)

2. *area\_ratio* : the ratio of *def\_area* to *box\_area*  
( $area\_ratio = def\_area / box\_area$ )

where *box\_area* is the area of the smallest rectangle enclosing the defect pattern.

3. *compactness* : the compactness of a pattern

$$((4 *area)/perimeter^2)$$

where perimeter is the length of the outline.

The probabilistic concept of moment has been used widely in pattern recognition as a practical method to extract features of the shape. Among the features of moment, the information useful for the inspection of the defects of cold mill strips are (a) the length of the longest axis, (b) the ratio of the longest axis to the shortest axis, and (c) the spread of a pattern. These features can be calculated from Eqs. (6)•(10).

4. *length of the longest and shortest axes of a pattern*

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j f(x, y) \tag{6}$$

$$a = 2\sqrt{2} \sqrt{(\mu_{20} + \mu_{02} + \sqrt{((\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2)})} : \text{longest} \tag{7}$$

$$b = 2\sqrt{2} \sqrt{(\mu_{20} + \mu_{02} - \sqrt{((\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2)})} : \text{shortest} \tag{8}$$

where  $f(x, y)$  is the function of gray level of an image and  $\mu_{ij}$  is the central moment of a pattern.

5. *ratio of the longest axis to the shortest axis of a pattern*

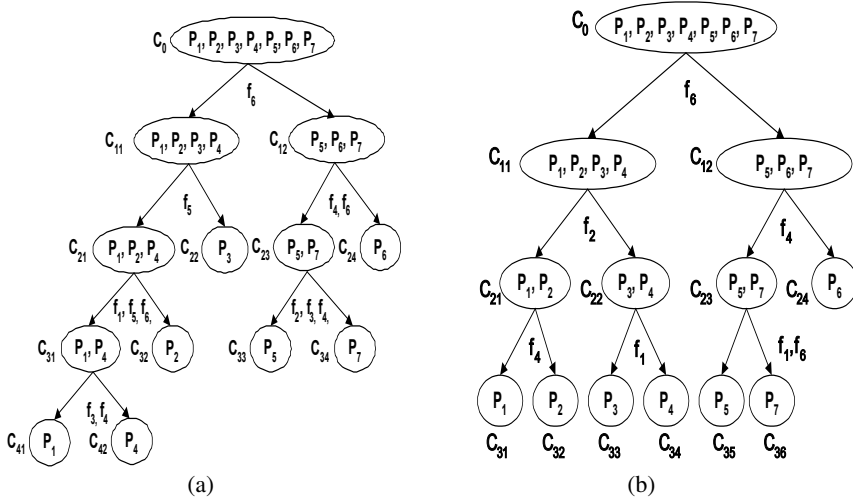
$$\text{axis\_ratio} = \frac{b}{a} \tag{9}$$

6. *Spread*

$$\text{spread} = \frac{\mu_{02} + \mu_{20}}{\mu_{00}^2} \tag{10}$$

### 3.2 Construction of the Binary Tree Recognizer

The data used in constructing the binary tree recognizer are the feature vectors extracted from the seven types of standard defect patterns. In constructing the binary tree using GA, the weights in Eq. (5),  $w_e$  and  $w_b$ , are set to 1. Fig. 3(a) shows the binary decision tree constructed from standard patterns by K-means algorithm. Fig. 3(b) represents the binary decision tree by GA. In Fig. 3,  $P_i$  is a type of pattern,  $f_k$  is a feature, and  $C_{mn}$  represents a class at each node.



Symbol	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>
Pattern	dull	oil	slip	dent	scale	dirt	Scratch

Symbol	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	f <sub>6</sub>
Feature	def_area	area_ratio	compactness	axis_ratio	spread	long_axis

**Fig. 3.** Binary decision tree constructed with standard patterns (a) Evaluation of feature subset by K-means algorithm (b) Evaluation of feature subset by GA

Table 1, 2 show the classification errors, balance coefficients, and the fitness values at each node. The classification errors in Table 1, 2 represent the number of patterns that leave their class when the patterns are divided into two groups at the node.

At each node constructed above, the final recognizer is made by training the BP neural network with the selected feature subset. The number of nodes in the input layer is set to the number of the selected features, and the number of nodes in the hidden layer is set to 10. By setting the number of nodes in the output layer to 2, the output layer represents the binary decision.



**Table 1.** The patterns and fitness at each node (K-means algorithm)

<i>Node</i>	<i>Pattern</i>	<i>Error/Patterns</i>	<i>Feature</i>	<i>Fitness</i>
$C_0$	$P_1P_2P_3P_4P_5P_6P_7$	0/38	$f_6$	0.5833
$C_{11}$	$P_1P_2P_3P_4$	0/15	$f_5$	0.4926
$C_{12}$	$P_5P_6P_7$	2/23	$f_4f_6$	0.2494
$C_{21}$	$P_1P_2P_4$	0/10	$f_1f_5f_6$	0.5858
$C_{23}$	$P_5P_7$	1/8	$f_2f_3f_6$	0.7795
$C_{31}$	$P_1P_4$	4/21	$f_3f_4$	0.1874

**Table 2.** The patterns and fitness at each node (GA)

<i>Node</i>	<i>Pattern</i>	<i>Error/Patterns</i>	<i>Feature</i>	<i>Fitness</i>
$C_0$	$P_1P_2P_3P_4P_5P_6P_7$	0/38	$f_6$	0.7180
$C_{11}$	$P_1P_2P_3P_4$	0/15	$f_2$	0.5677
$C_{12}$	$P_5P_6P_7$	2/23	$f_4$	0.2550
$C_{21}$	$P_1P_2$	0/8	$f_4$	0.5858
$C_{22}$	$P_3P_4$	0/7	$f_1$	0.7795
$C_{23}$	$P_5P_7$	3/21	$f_1f_6$	0.1874

Table 3 shows the results of recognizing the defect patterns of a cold mill strip using the binary tree recognizer.

In Table 3, the recognition rates of Dent and Slip are very low. However, Table 2 shows that the linear classification errors are zero at nodes  $C_0$ ,  $C_{11}$ , and  $C_{22}$  when constructing the binary decision tree. This means that the standard patterns of Dent and Slip are classified linearly. Because the least number of features that fit to classify standard patterns are selected, if the number of standard patterns is small, the recognizer becomes sensitive to noise.

**Table 3.** Recognition rate of each defect pattern

Patterns	<i>K-means algorithm</i>		<i>GA</i>	
	No. of recog. / No. of patterns	Recognition rate(%)	No. of recog. / No. of patterns	Recognition rate(%)
Dent	0/3	0	0/3	0
Dull	7/12	58.4	6/12	50
Oil drop	4/4	100	4/4	100
Slip	3/4	75	1/4	25
Dirt	2/2	100	2/2	100
Scale	16/22	72.7	19/22	86.3
Scratch	5/8	62.6	7/8	87.5
Total	37/55	67.2%	39/55	71%

## 4 Conclusions

In this paper, we used a binary decision tree classifier to recognize the defect patterns of a cold mill strip. We have used the cold mill strip of POSCO(Pohang Steel Company), which consists of 55 defect patterns. At each node of the binary tree, K-means and GA were used for the selection of the best feature subset and the linear decision function. There are two advantages of this method. One is that the construction of the binary decision tree and the selection of the best feature subset can be executed automatically for the given patterns. The other is that by designing the fitness function of GA properly, the decision tree can be obtained by considering the balance of the classes as well as the classification error.

In this experiment, GA is better than K-means algorithm in performance. But current performance is about 71% of recognition rate. Further studies should be made to design classifiers which have more generalization capabilities and feature extraction methods which are mutual helpful for the recognition of the defect pattern of a cold millstrip.

## References

1. Kim K.M.: Design of a Binary Decision Tree for Recognition of the Defect Patterns of Cold Mill Strip Using Generic Algorithm. Proc. AFSS. (1998) 208-212
2. Brill F., Brown D., Martin W.: Fast Genetic Selection of Features for Neural Network Classifiers. IEEE Trans. Neural Networks, 32 (1992) 324-328
3. Yao L.: Nonparametric Learning of Decision Regions via the Genetic Algorithm. IEEE Trans. Sys. Man Cybern., 26 (1996) 313-321
4. Mui J.K., Fu K.S.: Automated Classification of Nucleated Blood Cells using a Binary Tree Classifier. IEEE Trans. Pattern Analysis Machine Intelligence, 5 (1980) 429-443
5. Safavian S.R., Landgrebe D.: A Survey of Decision Tree Classifier Methodology. IEEE Trans. Sys. Man Cybern., 21 (1991) 660-674
6. Payne H.J., Meisel W.S.: An Algorithm for Constructing Optimal Binary Decision Trees. IEEE Trans. Computers, 26 (1997) 905-916
7. Swain P.H., Hauska H.: The Decision Tree Classifier: Design and Potential. IEEE Trans. Geosci. Elec., 15 (1997) 142-147
8. Jung S.W., Park G.T.: Design and Application of Binary Decision Tree using Genetic Algorithm. Journal of the KITE, 33(6) (1996) 1122-1130