Fuzzy Clustering Decision Tree for Classifying Working Wafers of Ion Implanter

Shih-Cheng Horng¹, Yu-Liang Hsiao²
Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan, R.O.C.
¹(schong@cyut.edu.tw), ²(s9727633@cyut.edu.tw)

Abstract - In this paper, we propose a fuzzy clustering decision tree (FCDT) for the classification problem with large number of classes and continuous attributes. A hierarchical clustering concept is introduced to achieve a finer fuzzy partition. The proposed clustering algorithm split the data set into leaf clusters using splitting attributes based on a separation matrix and fuzzy rules. The leaf clusters consisting of the data of more than one class will be further classified using the C4.5. We have successfully applied the FCDT to the classification problem of the working wafers in an ion implanter, and compared the classification results and the computation time with the existing software See5 and CART.

Keywords - Classification, fuzzy clustering, decision tree, C4.5, ion implanter.

I. INTRODUCTION

There exist numerous classification techniques for classification problems of continuous attributes such as the neural network approach [1]-[2], maximum-likelihood approach [3]-[4], fuzzy set theory based approach [5]-[6], decision tree [7]-[8]. Among them, the neural network approach is superior in the aspects of free data distribution and free data importance, however they are computationally expensive and produce variable results due to the random initial weights. The maximum-likelihood approach was the most widely used method in classifying remotely measurement data, however its performance was degraded when the target classes could not be adequately described by the statistical model. The fuzzy set theory based approach had been successfully applied to the pattern classification problem, however the computational complexity is raised when the number of classes as well as the number of attributes are large.

Decision trees are among the machine learning techniques that have the ability to handle complex problems by providing an understandable representation that is easier to interpret by producing logical rules of classification. The most widely used classification models are the entropy-based decision tree algorithms like ID3 [9] and C4.5 [10], which generate a multiway decision tree. The supervised learning in CART [11]-[12] and Quest (SLIQ) [13] algorithm uses the Gini index as a split measure to generate a binary decision tree. C4.5 is a very reliable classification technique for continuous attributes. Advantage of C4.5 is its high classification accuracy and free of data distribution. However, it is computationally intensive when both the size of the data set and the number of classes are large, because it treats each continuous value as a discrete one. Treating each continuous value as a discrete one will cause a large tree complexity, thus pruning after classification is a necessary step in C4.5. Thus, we can exploit this property to overcome the computational complexity of C4.5 while keeping its high classification accuracy. To fulfill this objective, we propose a separation matrix based clustering algorithm as a preprocessing step for C4.5. This clustering algorithm will classify the whole data set into a clustering tree and the classes in the leaf clusters will be classified by the C4.5. Because both the size and the number of classes of the leaf cluster are much smaller than the original data set, the computational complexity of C4.5 can be resolved.

We organize our paper in the following manner. In Section 2, we will present the fuzzy clustering decision tree. In Section 3, we will apply to an example and show the simulation results of FCDT. In Section 4, we will draw a conclusion.

II. THE FUZZY CLUSTERING DECISION TREE

The Separation Matrices Based Clustering Algorithm

2.1.1 Three-Sigma Limit Based Separation Matrices

The value of a continuous attribute in the data pattern of a class may spread over a range with certain probability distribution. Thus, the separability between two classes can be investigated through the degree of overlapping of the attribute-values. We let \( D(C_i, C_j) \) denote the separation index between classes \( C_i \) and \( C_j \) based on attribute \( k \) and define

\[
D(C_i, C_j)_k = \begin{cases} 
0, & \text{if } C_i \text{ and } C_j \text{ are separable based on attribute } k, \\
1, & \text{otherwise.} 
\end{cases}
\]

Clearly, \( D(C_i, C_j)_k = 1 \) and \( D(C_i, C_j)_k = D(C_j, C_i)_k \) for any attribute \( k \). The value of \( D(C_i, C_j)_k \) is computed using three-sigma limit [14] as described below. We let the random variable \( x^k_i \) denote the \( k \) th attribute of class \( C_i \), and let \( \mu^k_i \) and \( \sigma^k_i \) denote the mean and standard deviation of \( x^k_i \), respectively. We let the random variable \( x^k_j \) denote the \( k \) th attribute of class \( C_j \), and let \( \mu^k_j \) and \( \sigma^k_j \) denote the mean and standard deviation of \( x^k_j \), respectively. Without loss of generality, we can assume \( \mu^k_i < \mu^k_j \). If \( \mu^k_i - 3\sigma^k_i \) is larger than \( \mu^k_j + 3\sigma^k_j \), which implies the overlapping of the classes \( C_i \) and \( C_j \) on attribute \( k \)
will be very small. For example, the classes $C_1$ and $C_2$ are more likely to be separable as illustrated in Fig. 1; while in the case of Fig. 2, the two classes are not. Therefore $D(C_i,C_j)_k$ can be determined by the following:

$$D(C_i,C_j)_k = \begin{cases} 0, & \text{if } \mu_i^k + 3\sigma_i^k < \mu_j^k - 3\sigma_j^k, \\ 1, & \text{otherwise}. \end{cases} \quad (2)$$

2.1.2 Splitting Cluster Using Separation Matrices

Let $C_{r_0}$ denote the root cluster, which represents the whole data set. We define $[D(C_i,C_j)_k]_{h_1}$ as the separation matrix for all classes in $C_{r_0}$ based on attribute $k_i$. If $C_j$ belong to a disjoint block if $D(C_i,C_j)_k = 1$. The numbers of disjoint blocks represent the numbers of disjoint clusters that can be split from $C_{r_0}$ using attribute $k_i$. For example, the separation matrix $[D(C_i,C_j)_k]_{h_1}$ given in Fig. 3 consists of two disjoint clusters $\{C_1,C_2,C_3,C_4\}$ and $\{C_5,C_6,C_7,C_8\}$. The resulted clusters can be further split by other attributes.

Algorithm I: Build the clustering tree.

Step 0: Given the separation matrices of all attributes. Set $C_{r_0}$ as the root cluster and define the set of YSC $\{C_{r_0}\}$.

Step 1: For each cluster in YSC, obtain the corresponding splitting attribute $k$ based on (3). Using the obtained attribute to split the cluster, and put the resulting child clusters into YSC. Discard the clusters that had been split and the clusters that cannot be split using any attribute.

Step 2: If YSC = $\phi$, stop; otherwise return to Step 1.

2.1.3 The Choice of Attributes for Cluster Splitting

In general, for a given range of attribute values, more finer fuzzy partition is needed to classify a cluster with larger number of classes. Considering that any inaccurate cluster splitting will influence the accuracy of the subsequent cluster splitting along the tree path, we set the criteria for choosing the attribute to split a cluster as maximizing the number of classes and minimizing the variation of the number of classes in the resulted child clusters. We let $L_i$ and $n_{i-1}(C_r)$ denote the number of child clusters and the number of classes in the $i$th child cluster resulted from using attribute $k$ to split the cluster $C_r$, respectively. Then, the criteria for choosing the attribute for splitting $C_r$ is

$$\min_i \alpha_i(C_r) = \frac{1}{L_i} \left( \frac{1}{L_k} \right) \left( \frac{1}{L_k} \right) \frac{\sum_{i=1}^{k} \left( n_{i-1}(C_r) - \mu_i(C_r) \right)^2}{\sum_{i=1}^{k} \frac{1}{L_k}}$$

where $\mu_i(C_r) = \frac{\sum_{i=1}^{k} n_{i-1}(C_r) - \mu_i(C_r)}{L_k}$ denotes the variance of the number of classes, and $\mu_i(C_r) = \frac{\sum_{i=1}^{k} n_{i-1}(C_r)}{L_k}$ represents the average number of classes in the resulted child clusters. For example, for the separation matrices shown in Fig. 4 and 5, suppose that we use the attribute $k_1$ to split the cluster first, we obtain three child clusters. One consists of four classes, and the other two consist of three and one classes. The value of $\alpha_i(C_r)$ is 1.11. If we use attribute $k_2$ first, we will obtain four child clusters, and each child cluster contains two classes. The value of $\alpha_i(C_r)$ is 0.25. Since the value of $\alpha_i(C_r)$ is the smallest, we would choose $k_2$ to split the cluster.

$$\begin{array}{cccccccc}
C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
\end{array}$$

Fig. 3. A separation matrix example $[D(C_i,C_j)_k]_{h_1}$.

$$\begin{array}{cccccccc}
C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{array}$$

Fig. 4. The separation matrix $[D(C_i,C_j)_k]_{h_1}$.

$$\begin{array}{cccccccc}
C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
\end{array}$$

Fig. 5. The separation matrix $[D(C_i,C_j)_k]_{h_1}$.
the clustering tree as the leaf cluster (LC). Each LC may contain one class only or several classes, which cannot be split further using any attribute. To classify a new data pattern into a LC, we need to use the splitting attributes to construct the cluster splitting rules for each cluster based on the fuzzy rules [15]-[16] for single attribute.

Fig. 6. An example of using Algorithm I to build the clustering tree.

2.1.4 The Clustering Algorithm

The clustering algorithm consists of two parts: the training part and the classification part. The training part consists of three steps: (i) construction of the separation matrices for all attributes, (ii) determine the cluster splitting attribute and build the clustering tree, and (iii) throughout the clustering tree, generate the fuzzy if-then rules needed to classify a data pattern into proper child cluster based on a given set of training data patterns with known LCs.

a) The Fuzzy-Rule Generation Procedures

We let \( C_r \) denote a non-LC cluster and \( k \) denote the corresponding splitting attribute. We let \( x_s' \), \( s = 1, \ldots, g \), denote the \( k \) th attribute of \( g \) data patterns from \( M_j \) known child clusters, \( C C r_j 1 \ldots, C C r_j K \). These \( g \) data patterns form the training data set for splitting \( C r_j \). The fuzzy rules \( R_j (C r_j) \) for splitting cluster \( C r_j \) are described below. For \( i = 1, \ldots, K \), where \( K \) denotes the number of fuzzy partitioned intervals on the range of the \( k \) th attribute values, if \( x_s' \) is \( A_k \), then the \( x_s' \) belongs to \( C C r_j \) with \( C F_i \), where \( A_k \) is the \( i \) th partitioned fuzzy interval, \( C C r_j \) is the consequent, and \( C F_i \) is the grade of certainty of rule \( R_j (C r_j) \).

What need be determined in the above rule are \( C C r_j \) and \( C F_i \), and the procedures for determining them are called fuzzy rules generation procedures for splitting one cluster. Let \( A_k \) be characterized by the fuzzy membership function \( f_k (x_s') \). Then, \( f_k (x_s') \) can be considered as the grade of compatibility of \( x_s' \) with respect to \( A_k \). We define \( \beta (R_j (C r_j)) = \sum_{i \in C C r_j} f_k (x_s') \) as the sum of grade of compatibility of child cluster \( C C r_j \) with respect to \( A_k \).

Algorithm II: Generation of the \( K \) fuzzy rules.

Step 0: Given \( g \) training data patterns \( x_s' \), \( s = 1, \ldots, g \), with known child clusters \( C C r_j \), \( l = 1, \ldots, M_j \) of the to-be-split cluster \( C r_j \) and the corresponding splitting attribute \( k \). Set \( i = 1 \).

Step 1: Calculate the sum of grade of compatibility of child cluster \( C C r_j \), \( l = 1, \ldots, M_j \), with respect to \( A_k \).

Step 2: Find the child cluster \( C C r_j \) such that

\[
\beta (R_j (C r_j)) = \max \{ \beta (R_j (C r_j)) \}
\]

then \( C C r_j \) is the consequent \( C C r_j \) in rule \( R_j (C r_j) \).

Step 3: Determine \( C F_i \) by

\[
C F_i = \frac{\beta (R_j (C r_j)) - \beta (R_j (C r_j))}{\sum_{i = 1}^{M_j} \beta (R_j (C r_j))} (M_j - 1)
\]

where \( \beta (R_j (C r_j)) = \sum_{i \in C C r_j} f_k (x_s') \) is the average of the sum of grade of compatibility of the rest of child clusters with respect to \( A_k \).

Step 4: If \( i = K \), stop; else, set \( i = i + 1 \), and return to Step 1.

b) Training Part of the Clustering Algorithm

Combining the construction of separation matrices, determination of the splitting attributes, building of the clustering tree and the above fuzzy rule generation procedures, the training procedures of the clustering algorithm using the training data set are as below.

Algorithm III: Training of the clustering algorithm.

Step 0: Given a set of training data patterns with known classes; compute \( \mu_i \) and \( \sigma_i \) of each class \( C_i \) and each attribute \( k \); compute the separation matrices \( [D(C_i, C_j)] \) based on (2) for each attribute \( k \).

Step 1: Apply Algorithm I to obtain the splitting attributes and build the clustering tree.

Step 2: Use Algorithm II to generate the fuzzy rules for each cluster in the clustering tree.

c) Classification Part of the Clustering Algorithm

Once the fuzzy rules for splitting the clusters in the clustering tree are generated, we can determine the child cluster to which the new data pattern belongs at each cluster based on a fuzzy reasoning method. Let the new data pattern be \( x' \) and let \( x_s' \) be the \( k \) th attribute of \( x' \) corresponding to the splitting attribute \( k \) at cluster \( C r_j \). We define \( \gamma (C r_j) \), the weighting grade of certainty of \( x_s' \) with respect to the child cluster \( C r_j \), as the sum of the multiplication of the grade of compatibility of \( x_s' \) with respect to \( A_k \) and the grade of certainty of rule \( R_j (C r_j) \) over all \( K \) trained rules whose consequent are \( C C r_j \). \( \gamma (C r_j) \) can be expressed as

\[
\gamma (C r_j) = \sum_{R_j (C r_j) \in C r_j \subseteq C C r_j} f_k (x_s') \cdot C F_i \]

Algorithm IV: Classification of the clustering algorithm.
Step 0: Given a new data pattern \( x' = (x_1', ..., x_n') \), where \( n \) denotes the total number of attributes; set Present Cluster \( (PCr)_n = C_{n0} \).
Step 1: The child cluster \( C_{Cr_{k'}} \), with respect to which the weighting grade of certainty of \( x'_i \) is maximum, is the concluded cluster of \( x' \).
Step 2: Use \( x'_i \), where \( k \) corresponds to the attribute used for splitting the \( PCr \), and Step 1 to classify \( x' \) into a child cluster of \( PCr \), we denote this child cluster by \( CPCr \). If the \( CPCr \) is not a LC, set \( PCr = CPCr \) and repeat this step; otherwise, stop.

2.2 C4.5 for Classifying the Leaf Cluster (LC)

The Leaf Clusters (LCs) resulted from the training part of the separation matrix based clustering algorithm may consist of one or more classes. Since the number of classes and the size of the corresponding data set in each LC should be much smaller than \( C_{n0} \), it will be computationally much easier to apply C4.5 to classify the LCs. Thus, our clustering algorithm help reduce the computational complexity of C4.5. In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.

Similar to the proposed clustering algorithm, C4.5 also consists of training and classification parts. The training part of C4.5 is to build a classification tree and the splitting rules in each node. C4.5 builds decision trees from a set of training data using the concept of information entropy. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. C4.5 goes back through the tree once it’s been created and attempts to remove branches that do not help by replacing them with leaf nodes.

2.3 Classification of a New Data Pattern

Once the training part of the FCDT, which combines the training parts of the clustering algorithm and C4.5, is completed, we are ready to use the classification procedures of both clustering algorithm and C4.5 to classify a new data pattern.

III. TEST RESULTS

3.1 Application Example

Ion implanter is a bottleneck machine in the semiconductor manufacturing process because of its expensiveness, and ion implantation is a critical operation to the yield rate [17]. In a typical semiconductor foundry, there are several tens to hundreds of recipes for wafer fabrication each day. We can view a recipe as a class. Thus, the recipe classification for ion implantation is of course a classification problem with large number of classes, which is especially suitable for the application of the proposed technique.

In general, there are quite a few attributes that can be measured from the ion implanter; however, not all attributes are helpful in classification. According to the domain knowledge, the following ten attributes, \( k_1, ..., k_{10} \), are recommended: filament voltage, filament current, extraction electrode voltage, extraction electrode current, acceleration/deceleration voltage, magnetic field strength, high voltage power supply current, beam current, beamline pressure, and chamber pressure, respectively. These attributes cover the four subsystems of the ion implanter. A data set of 36-recipe case, and each recipe consists of a thousand to 10,000 wafers are supported from a local world renowned foundry. We use them to test the classification accuracy of the proposed FCDT.

3.2 Test Results of FCDT

For all the measured data patterns, we randomly divide them in wafer base into ten parts. We take 9 parts as training data set and 1 part as test data set. The number of fuzzy partitioned intervals \( K = 12 \) and a triangle nonnegative membership function for \( j^k \) in Algorithm II. Applying Algorithm III to the training data set, the resulting clustering tree and the splitting attributes are shown in Fig. 7, where each cluster is denoted by a block, and the classes contained in a cluster are shown inside the parenthesis in each block. The attribute used for splitting each cluster is indicated at the outgoing branch in the clustering tree. The corresponding fuzzy rules for each splitting attribute are also obtained. There are eight LCs, and each LC consists of more than one class except for LC\(_3\) and LC\(_5\). Subsequently, we apply C4.5 to the other six LCs and build the classification tree and splitting rules for each LC. We then use the part of test data to test the trained FCDT using Algorithm IV of the clustering algorithm and the classification tree and splitting rules of C4.5. Repeating this process for ten times by circulating the training data set and test data set, Table I shows the resulting 10-fold cross validation classification error rates of all recipes. We also indicate the classification error rates using the software See5 [18] and CART [11] in Table I. The resulting 10-fold cross validation for the sum of classification error rates and the corresponding training times are shown in Table II. The 10-fold cross validation for the sum of classification error rates of the proposed FCDT is around 1.50%; while the sum of classification error rates using See5 and CART are 3.02% and 6.79%, which are 101% and 353% more than that of FCDT, respectively. Thus, FCDT obtains a very successful classification result.
In this paper, we have proposed a fuzzy clustering decision tree (FCDT) for the classification problem with large number of classes and attributes. This technique can effectively reduce the computational complexity because of the reduction of the size of fuzzy rule set. It can also increase the classification rate due to the finer fuzzy partition achieved in the LC than in the root cluster. As a demonstrated example, we have successfully applied the proposed classifier on the ion implantation classification problem.

IV. CONCLUSION

ACKNOWLEDGMENT

This research work is supported in part by the National Science Council in Taiwan, R.O.C., under grant NSC98-2221-E-324-039-.

REFERENCES