Designing Cost-sensitive Fuzzy Classification Systems Using Rule-weight

Mansoor Zolghadri Jahromi[1], Mohammad Reza Moosavi[2]
School of Electrical and Computer Engineering, Shiraz University,
Shiraz, Iran
[1]zjahromi@shirazu.ac.ir, [2]mmoosavi@cse.shirazu.ac.ir

***Abstract*— In the field of pattern classification, we often encounter problems that class-to-class misclassification costs are not the same. For example, in the medical domain, misclassifying a patient as normal is often much more costly than misclassifying a normal as patient. Our aim in this paper is to propose a method of designing fuzzy rule-based classification systems to tackle this problem. We use rule-weight as a simple mechanism to tune the rule-base. Assuming that class-to-class misclassification costs are known, we propose a learning algorithm that attempts to minimize the total cost of the classifier on train data (i.e., instead of minimizing the error-rate). Using a number of UCI datasets we show that the method is quite effective in reducing the average cost of the classifier on test data.**

Keywords- Fuzzy Classification Systems; Cost Sensitive Classification; Rule Weight; Data Mining

I. INTRODUCTION

A Fuzzy Rule-Based Classification System (FRBCS) is a special case of fuzzy modeling where the output of the system is crisp and discrete. Basically, the design of a FRBCS consists of finding a compact set of fuzzy if-then classification rules to be able to model the input-output behavior of the system. The information available about the behavior of the system is assumed to be a set of input-output example pairs (i.e., a number of pre-labeled classification examples).

The most challenging problem in designing FRBCSs is the construction of rule-base for a specific problem. Many approaches have been proposed to construct the rule-base from numerical data. These include heuristic approaches [1, 2], neuro-fuzzy techniques [3-5], clustering methods [6-8], genetic algorithms [9-12] and data mining techniques [13-15].

One main advantage of fuzzy rule-based systems in classification problems is their interpretability. Using linguistic labels in the antecedent of the fuzzy rules makes them very understandable, which is the main characteristic of this type of classifier.

There are many classification problems where class-to-class misclassification costs are different. For example, in medical diagnosis of cancer, diagnosing malignant tumors as benign and hence treating a cancer patient as healthy could be much more costly than interpreting benign tumors as malignant.

Cost-sensitive learning first introduced by Elkan [16] has shown to be an effective technique for incorporating the different misclassification costs into the classification process [17-21].

A pattern classification problem can be easily reformulated as a cost minimization problem. In [22], the concept of instance weight is introduced for each training pattern in order to handle the cost-sensitive problems. The weight of an input pattern represents the average cost of misclassifying that pattern. Fuzzy if–then rules are generated by considering the weights as well as the compatibility of training patterns. A rule-weight learning method based on Reward and Punishment is also proposed to tune the weight of the rules.

In this paper, we assume that for the problem in hand a cost matrix C giving class-to-class misclassification costs is available. We assume that the cost of misclassifying an instance depends on its actual and predicted classes. Each element c_{ij} of this matrix gives the cost of classifying a pattern from class i in class j (c_{ij}=0 if i=j). This is slightly different from the scheme that assumes that the misclassification cost of an instance depends only on its actual class [16, 20]. The design of the classifier is then viewed as a cost minimization problem.

For a specific cost-sensitive problem, an initial rule-base is constructed using one of the methods proposed in the literature [22]. The initial rule-base is then tuned to the problem in hand by assigning a weight to each fuzzy rule in the constructed rule-base. The novelty of our method is in the rule-weight learning algorithm that we propose. The proposed algorithm uses the cost matrix to directly minimize the total misclassification cost of the classifier on training data. In this process, the size of the rule-base is reduced by assigning zero weight to redundant rules, which improves the interpretability of the final rule-base. Using a number of datasets from UCI ML repository, we show that the scheme is quite effective in constructing a compact rule-base for cost-sensitive problems.

The rest of this paper is organized as follows. In Section II, the structure of a fuzzy classification system is introduced. In Section III, a method of constructing the rule-base for conventional (i.e., not cost sensitive) problems is discussed. In Section IV, a method of constructing rule-base for cost-sensitive problems is presented. In Section V, the proposed method of rule-weight learning is presented. In Section VI, the simulation results are presented. Section VII concludes this paper.

II. FUZZY RULE-BASED CLASSIFICATION SYSTEMS

A fuzzy rule-based classification system is composed of three main conceptual components: database, rule-base, and reasoning method. The database describes the semantic of fuzzy sets associated to linguistic labels. Each rule in the rule-base specifies a subspace of pattern space using the fuzzy sets in the antecedent part of the rule. The reasoning method provides a mechanism to classify a pattern using the information from the rule-base and database.

Different rule types have been used for pattern classification problems [23]. We use fuzzy rules of the following type for an n-dimensional problem:

$$\text{Rule } R_j\text{: If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then class } h \text{ with } CF_j,\quad j=1, 2, \ldots, N \tag{1}$$

where $X=[x_1, x_2, \ldots, x_n]$ is the input feature vector, $h \in \{C_1, C_2 \ldots, C_M\}$ is the label of the consequent class, A_{jk} is the fuzzy set associated to x_k , CF_j is the certainty grade (i.e., rule weight) of rule R_j and N is the number of fuzzy rules in the rule-base.

In order to classify an input query pattern $X_t = [x_{t1}, x_{t2}, \ldots, x_{tn}]$, the degree of compatibility of the pattern with each rule is calculated (i.e., using a T-norm to model the "and" connectives in the rule antecedent). In case of using product as T-norm, the compatibility grade of rule R_j with the input pattern X_t can be calculated as:

$$\mu_j(X_t) = \prod_{i=1}^{n} \mu_{A_{ji}}(x_{ti}) \tag{2}$$

Using single winner reasoning method, an input query pattern X_t is classified according to the consequent class of the winner rule R_w. With the rules of form (1), the winner rule R_w is identified as:

$$w = \arg\max_{1 \le j \le N} \{\mu_j(X_t).CF_j\} \tag{3}$$

III. RULE-BASE CONSTRUCTION

For an M-class problem in an n-dimensional feature space, assume that m labeled patterns of the form $X_p=[x_{p1}, x_{p2}, \ldots, x_{pn}]$, $p=1, 2, \ldots, m$ are given. A simple approach for generating fuzzy rules is to partition the domain interval of each input attribute using a pre-specified number of fuzzy sets (i.e., grid partitioning). Some examples of this partitioning (using triangular membership functions) are shown in Fig. 1.

Given a partitioning of pattern space, one approach is to consider all possible combination of the antecedents to generate the fuzzy rules. The selection of the consequent class for an antecedent combination (i.e., a fuzzy rule) can be easily expressed in terms of confidence of an association rule from the field of data mining [24]. A fuzzy classification rule can be viewed as an association rule of the form $A_j \Rightarrow class\ C_j$ where, A_j is a multi-dimensional fuzzy set representing the antecedent conditions and C_j is a class label. Confidence of a fuzzy association rule R_j is defined as [15]:

$$C(A_j \Rightarrow class\ C_j) = \frac{\sum_{X_p \in class\ C_j} \mu_j(X_p)}{\sum_{p=1}^{m} \mu_j(X_p)} \tag{4}$$

where $\mu_j(X_p)$ is the compatibility grade of pattern X_p with the antecedent of the rule R_j, m is the number of training patterns, and C_j is a class label.

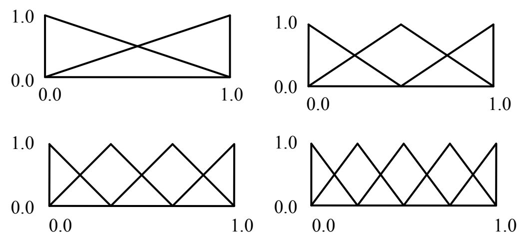

Figure 1. Different partitioning of each feature axis.

A common approach for identifying the consequent class C_q of an antecedent combination A_j is to specify the class with maximum confidence as the consequent class. This can be expressed as:

$$q = \arg\max_{1 \le h \le M} \left\{ C(A_j \Rightarrow Class\ C_h) \right\} \tag{5}$$

The problem with grid partitioning is that an appropriate partitioning of each attribute is not usually known. One solution for this is to simultaneously consider different partitioning of an attribute (see Fig. 1). That is, for each attribute, a pre-specified number of fuzzy sets (for example 14, as shown in Fig. 1) can be used when generating a fuzzy rule. The problem is that for an n-dimensional problem, 14^n antecedent combinations should be considered. It is impractical to consider such a huge number of antecedent combinations when dealing with high dimensional problems.

One solution for the above problem is presented in [15] by adding the fuzzy set "don't care" to each attribute. The membership function of this fuzzy set is defined as $\mu_{don't\ care}(x) = 1$ for all values of x. The trick is not to consider all antecedent combinations (which is now 15^n) and only short fuzzy rules having a limited number of antecedent conditions (excluding don't care) are generated as candidate rules.

The number of candidate rules generated with the above scheme can still be quite large for many problems. A compact rule-base can be constructed in the following manner. The generated candidate rules are divided into M groups according to their consequent classes. The candidate rules in each group are sorted in descending order of an evaluation criterion. A rule-base is constructed by choosing Q fuzzy rules from each class (i.e., $M \times Q$ fuzzy rules in total). Among many heuristic rule evaluation measures presented in the literature [25], we use the measure presented in [10]. The

 ISBN: 978-1-61208-162-5

evaluation of rule R_j (i.e., $A_j \Rightarrow class\ C_j$) with this measure can be expressed as:

$$e(R_j) = \sum_{X_p \in Class\, C_j} \mu_j(X_p) - \sum_{X_p \notin Class\, C_j} \mu_j(X_p) \qquad (6)$$

IV. COST-SENSITIVE FUZZY CLASSIFICATION SYSTEMS

For an M-class problem, assume that an $M\times M$ cost matrix C giving class-to-class misclassification costs is given. In this section, we extend the rule-base construction method of the previous section for the case of cost-sensitive problems. For this purpose, we assign a weight to each training example. The weight assigned to each training example is the average cost of classifying that example. The cost matrix C can be used to calculate the weight w_p a training example X_p (from class i) as:

$$w_p = \frac{1}{M}\sum_{j=1}^{M} c_{i,j} \qquad (7)$$

where, $c_{i,j}$ denotes the cost of classifying an instance of class i in class j. The weight assigned to a training example can be viewed as the importance of that pattern in the classification process. Using the weights of the training examples, the confidence of an association rule (4) can be easily modified to:

$$C(A_j \Rightarrow class\ C_j) = \frac{\sum_{X_p \in class\ C_j} w_p.\mu_j(X_p)}{\sum_{p=1}^{m} w_p.\mu_j(X_p)} \qquad (8)$$

The rule evaluation metric (6) is modified to accommodate the weights assigned to training examples:

$$e(R_j) = \sum_{X_p \in Class\, C_j} w_p.\mu_j(X_p) - \sum_{X_p \notin Class\, C_j} w_p.\mu_j(X_p) \qquad (9)$$

It must be noted that equations (8) and (9) cover the special case of cost-insensitive problems (i.e., w_p=1, p=1,2,..,m). In short, the rule generation process discussed in Section III can be used to construct a rule-base for a cost-sensitive problem. For this purpose, equations (4) and (6) are replaced by (8) and (9), respectively.

V. RULE-WEIGHT LEARNING ALGORITHM

For the problem in hand, assume that a rule-base consisting of N fuzzy classification rules $\{R_j, j=1, 2, ..., N\}$ is constructed using the method discussed in the previous section. Our aim in this section is to propose a rule-weight learning algorithm that attempts to minimize the total cost misclassification cost of the rule-base on train data. For this purpose, we make use of the rule-weight learning algorithm that was proposed in [26], which attempts to minimize the error-rate of the classifier on training data. In this section, we propose an extended version of this algorithm to cover case-sensitive problems. The proposed algorithm attempts to minimize the total misclassification cost of the constructed rule-base on the training data.

In its basic form, the proposed algorithm is a hill-climbing search method. The algorithm starts with an initial solution to the problem (i.e., $\{CF_k = 1, k = 1, 2, \ldots, N\}$) and attempts to improve the solution by adjusting the weight of each rule in turn (to reduce the total cost on train data). The basic component of the learning scheme is an algorithm (denoted as *best-weight*) that provides the answer to the following question: *"What is the optimal weight of a rule (i.e., R_k) assuming that the weights of all other rules are given and fixed?"*

The weight found by *best-weight* is optimal in the sense that it results in minimum total misclassification cost on training data. In this way, the overall learning algorithm consists of visiting each rule in turn to adjust its weight. It must be noted that the weight specified for a rule is optimal if the weights of other rules in the rule-base remain fixed. That is why the second pass and subsequent passes over the rules can reduce the cost on train data. In experiments, as a mechanism to prevent overfitting, we stop the search after a fixed number of passes over all rules [26].

To illustrate how the *best-weight* algorithm finds the optimal weight of a rule, consider rule R_k for optimization.

$$\text{Rule } R_k\text{: If } x_1 \text{ is } A_{k1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{kn} \text{ then class } T \text{ with } CF_k \qquad (10)$$

To calculate the optimal weight of rule R_k (i.e., CF_k), the rule is first removed from the rule-base by setting its weight to zero (CF_k=0). In the next step, the predicted class of all the training patterns will be found and stored (without rule R_k in the rule-base). Then, the score S of each training data X_t in covering subspace of rule R_k (i.e., $\mu_k(X_t) \neq 0$) is calculated using the following definition of score:

$$S(X_t) = \frac{\max_{1\le j\le N}\{CF_j.\mu_j(X_t) \mid R_j \neq R_k\}}{\mu_k(X_t)} \qquad (11)$$

where, $\mu_k(X_t)$ denotes the compatibility grade of pattern X_t with rule R_k . For a pattern X_t having score $S(X_t)=a$, if we choose $CF_k > a$, the pattern X_t will be classified by rule R_k (i.e., as class T) since rule R_k will be the winner rule (having maximum weighted compatibility with pattern X_t). In case we choose $CF_k < a$, the pattern will be classified as if we don't have rule R_k in the rule-base (we have already stored the predicted class in previous step).

For a specific value of CF_k, the predicted class of X_t (with $S(X_t)=a$) for the two interval of $CF_k < a$ and $CF_k > a$ are known. As we know the true class of X_t, we can easily calculate the cost of classifying X_t for $CF_k < a$ and $CF_k > a$. For a training pattern X_t, assume that L is the true class, P is the predicted class for $CF_k < a$, and T is the predicted class for $CF_k > a$. Then, the cost of classifying X_t for $CF_k < a$ and $CF_k > a$ can be expressed as:

$$Cost(X_t) = \begin{cases} C_{T,P} & if \ CF_k < a \\ C_{T,L} & if \ CF_k > a \end{cases} \quad (12)$$

where, $C_{I,J}$ is used to represent the cost of classifying an instance of class I in class J.

Having the relation between a certain value of CF_k and the corresponding total cost of training data, the best value of CF_k can be easily found by sorting the patterns in ascending order of their scores (i.e., $S(X_1) < S(X_2) < .... < S(X_n)$). Considering any value of CF_k between $S(X_i)$ and $S(X_{i+1})$, the first i patterns will be classified as class T and the rest of the patterns will be classified as if rule R_k is not in rule-base. In this way, $n+1$ different values of CF_k should be examined to find its best value. The *best-weight* algorithm for calculating the best weight of a rule is given in Fig. 2.

The algorithm starts by finding the predicted class of each pattern when the rule is removed from the rule-base. The patterns are then sorted in ascending order of their scores. For a rule having n training pattern in its covering space, the algorithm of Fig. 2 examines $n+1$ values to find the best weight (*best-CF*) for the rule. The first and last values are "zero" and "*last+ε*", respectively (last is the score of last pattern in the ranked list and ε is a very small positive number). The rest are examined in the middle of two successive scores.

VI. EXPERIMENTAL RESULTS

In order to assess the performance of the proposed method, we used four data sets available from UCI ML repository. Some statistics of these datasets are shown in Table I.

To construct an initial rule-base for a specific problem, we used the method of Section III to generate rules of *length*≤2. The candidate rules were the grouped based on their consequent classes. The rules in each group were then sorted according to the rule evaluation metric. An initial rule-base was constructed by choosing a certain number of best rules from each group. The proposed rule-weight learning algorithm was used to optimize the rule-base by passing 4 iterations over all rules.

We used 10-times 10-fold cross validation technique to assess the generalization ability of the proposed method. In each fold, 90% of the data were use to construct the rule-base. The proposed rule-weight learning algorithm was then used to specify the weights of all rules in the rule-base. The performance on test data was measured by calculating the average cost per example (CPE).

For the purpose of experiments we assumed a cost matrix using the number of instances in each class (i.e., class proportionate cost). The misclassification cost of predicting an instance of class i in class j is assumed to be:

$$C_{ij} = \frac{\text{(total no. of patterns of class j)}}{\text{(total no. of patterns of class i)}} \quad (13)$$

This cost matrix assumes that misclassification of the minority class (with a small number of training patterns) is more costly than majority class. In Tables II and III we give the cost matrix used for each dataset, which is based on equation (13).

It must be noted that the misclassification costs in most medical problems depends strongly on the domain, and particularly the anticipated consequences of the misclassification. This is not directly related to the class proportions. These cost matrices are used as examples (i.e., they don't represent the actual misclassification costs) to evaluate the proposed rule-weight learning algorithm.

```
Inputs: training patterns in the covering subspace of the rule and true class of each
pattern {(Xt, true-class(Xt)), t=1,2,…,n}

Output: the best weight for the rule (best-CF) assuming that the weights of all other
rules are fixed

CF = 0    (i.e. remove the rule from the rule-base)
for each training pattern, Xi
      Calculate and memorize the predicted class of Xi
      Calculate and memorize S(Xi) using eq. 11
rank the patterns in ascending order of their scores in a list

#assume that Xk and Xk+1 are two successive patterns in the list
#also assume that Xlast is the last pattern in the list and ε is a small positive #number

for each value of CF (i.e. CF=0, CF=(Score(Xk)+Score(Xk+1))/2, CF=Score(X last)+ε)
      Calculate and memorize total misclassification cost corresponding to the specified
      value of CF

best_CF = CF with minimum total misclassification cost

return best-CF
```

Figure 2. Best-Weight Algorithm for finding the best weight of a rule

In Table IV, we report the CPE for the initial rule-base (i.e., before rule-weighting) and after applying the rule-weighting algorithm of Section IV. As seen, our rule-weighting algorithm has significantly reduced the CPE on test data for all datasets used in our experiments.

In order to assess the performance of our method in comparison with other methods proposed in the literature, in Table V, we report the results of the method proposed in [22] to handle cost sensitive problems. The cost matrixes used to produce the results of this Table is the same as Table IV (i.e., class proportionate cost matrix (13)).

Comparing the results of Tables IV and V, we observe that our proposed rule-weighting algorithm outperforms the method proposed in [22] by achieving lower value of CPE on test data for all datasets used in experiments, which was the primary goal of the algorithm.

TABLE I. SOME STATISTICS OF THE DATASETS USED IN EXPERIMENTS.

Dataset	No. Of features	No. of instances	No. of classes	No. of instances per class
Thyroid	5	215	3	35, 30, 150
Pima	8	768	2	500, 268
Bupa	6	345	2	145, 200
Breast cancer	30	569	2	357, 212

TABLE II. CLASS-PROPORTIONAL COST MATRIX FOR THYROID DATASET.

	hyper-thyroidism	hypo-thyroidism	normal
hyperthyroidism	0	0.86	4.28
hypothyroidism	1.17	0	5
normal	0.23	0.2	0

TABLE III. CLASS-PROPORTIONAL COST MATRICES.

Pima	***tested_negative***	***tested_positive***
tested_negative	0	0.536
tested_positive	1.87	0
Bupa	***drinks<5***	***drinks>5***
drinks<5	0	1.38
drinks>5	0.73	0
Breast cancer	***Benign***	***malignant***
benign	0	0.594
malignant	1.68	0

TABLE IV. THE CPE ON TRAIN AND TEST DATA FOR VARIOUS DATASETS USING OUR PROPOSED METHOD.

Dataset	Train data		Test data	
	Before rule-weighting	After rule-weighting	Before rule-weighting	After rule-weighting
Thyroid	1.24	0.03	1.31	0.12
Pima	1.42	0.42	1.43	0.54
Bupa	0.58	0.26	0.59	0.36
Breast cancer	0.12	0.03	0.13	0.06

TABLE V. THE CPE ON TRAIN AND TEST DATA FOR VARIOUS DATASETS USING THE METHOD PROPOSED IN [22].

Dataset	Train data		Test data	
	Before rule-weighting	After rule-weighting	Before rule-weighting	After rule-weighting
Thyroid	0.25	0.15	0.25	0.2
Pima	0.96	0.79	0.97	0.8
Bupa	0.57	0.4	0.58	0.42
Breast cancer	0.52	0.24	0.55	0.24

In Table VI, we report on average number of rules in the final rule-base using our method. As seen, the number of rules in the final rule-base is much smaller than initial rule-base. This is due to the fact that our algorithm removes the redundant rules by setting their weights to zero. This is important since the interpretability and efficiency of the rule-base is improved.

TABLE VI. AVERAGE NUMBER OF RULES IN THE FINAL RULE-BASE USING THE PROPOSED METHOD.

Dataset	Before rule-weighting	After rule-weighting
Thyroid	99	4.53
Pima	66	10.5
Bupa	66	11.9667
Breast cancer	50	6.2

VII. CONCLUSIONS

In this paper, a cost-sensitive learning algorithm was proposed to tune a fuzzy classification system by specifying the weights of fuzzy rules. The learning algorithm makes use of the cost matrix giving class-to-class misclassification costs to minimize the total cost on train data.

Using a number of real-life datasets, we showed that the scheme is quite effective in reducing the average cost of the classifier on test data. Another advantage of the proposed method is that redundant rules are removed during the learning process. This feature is very useful since the final rule-base is better in terms of interpretability and classification speed.

Since the proposed learning method attempts to minimize the classification cost of the classifier on training data, obviously, this can cause the classifier to overfit the training data. The main cause for this is that the learning algorithm does not have a mechanism to cope with noisy training examples (i.e., those in contradiction with the rest of training patterns). A mechanism is needed to deal with this issue.

REFERENCES

[1] S. Abe and M. S. Lan, “A method for fuzzy rules extraction directly from numerical data and its application to pattern classification,” IEEE Trans. on Fuzzy Systems, vol. 3, Feb. 1995, pp. 18-28.

[2] H. Ishibuchi, K. Nozaki and H. Tanaka, “Distributed representation of fuzzy rules and its application to pattern classification,” Fuzzy Sets and Systems, vol. 52, Nov. 1992, pp. 21-32 .

[3] S. Mitra and L. I. Kuncheva, “Improving classification performance using fuzzy MLP and two-level selective partitioning of the feature space,” Fuzzy Sets and Systems, vol. 70, Feb. 1995, pp. 1-13.

[4] D. Nauck and R. Kruse, “A neuro-fuzzy method to learn fuzzy classification rules from data,” Fuzzy Sets and Systems, vol. 89, Aug. 1997, pp. 277-288.

[5] I. Gadaras and L. Mikhailova, “An interpretable fuzzy rule-based classification methodology for medical diagnosis,” Artificial Intelligence in Medicine, vol. 47, Sep. 2009, pp. 25-41.

[6] J. A. Roubos, M. Setnes, and J. Abonyi, “Learning fuzzy classification rules from labeled data,” Information Sciences, vol. 150, Mar. 2003, pp. 77-93.

[7] J. Abonyi and F. Szeifert, “Supervised fuzzy clustering for the identification of fuzzy classifiers,” Pattern Recognition Letters, vol. 24, Feb. 2003, pp. 2195–2207.

[8] P. Pulkkinen and H. Koivisto, “Identification of interpretable and accurate fuzzy classifiers and function estimators with hybrid methods,” Applied Soft Computing, vol. 7, Mar. 2007, pp. 520-533.

[9] J. Casillas, O. Cordón, and M. J. Del Jesus, F. Herrera, “Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems,” Information Sciences, vol. 136, Aug. 2001, pp. 135-157.

[10] A. Gonzalez and R. Perez, “SLAVE: A genetic learning system based on an iterative approach,” IEEE Trans. on Fuzzy Systems, vol. 7, Apr. 1999, pp. 176-191.

[11] H. Ishibuchi, T. Nakashima, and T. Murata, “Three-objective genetics-based machine learning for linguistic rule extraction,” Information Sciences, vol. 136, Aug. 2001, pp. 109-133.

[12] L. Sánchez, I. Couso, J. A. Corrales, O. Cordón, M. J. Del Jesus, and F. Herrera, “Combining GP operators with SA search to evolve fuzzy rule based classifiers,” Information Sciences, vol. 136, Aug. 2001, pp. 175-191.

[13] Y. Chung Hu and G. Hshiung Tzeng, “Elicitation of classification rules by fuzzy data mining,” Engineering Applications of Artificial Intelligence, vol. 16, Oct. 2003, pp. 709-716.

[14] M. De Cock, C. Cornelis, and E. E. Kerre, “Elicitation of fuzzy association rules from positive and negative examples,” Fuzzy Sets and Systems, vol. 149, Jan. 2005, pp. 73-85.

[15] H. Ishibuchi and T. Yamamoto, “Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining,” Fuzzy Sets and Systems, vol. 141, Jan. 2004, pp. 59-88.

[16] C. Elkan, “The foundations of cost-sensitive learning,” Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 01), June 2001, pp. 973-978.

[17] T. Nakashima, Y. Yokota, H. Ishibuchi, and G. Schaefer, “A cost-based fuzzy system for pattern classification with class importance,” Artificial Life and Robotics, vol. 12, Sep. 2008, pp. 43-46.

[18] J. Zheng, “Cost-sensitive boosting neural networks for software defect prediction,” Expert Systems with Applications, vol. 37, June 2010, pp. 4537–4543.

[19] S. Viaene and G. Dedene, “Cost-sensitive learning and decision making revisited,” European Journal of Operation Research, vol. 166, Oct. 2005, pp. 212–220.

[20] L. Li, M. Chen, H. Wang, and H. Li, “CoSFuC: A Cost Sensitive Fuzzy Clustering Approach for Medical Prediction,” Proc. of Fifth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 08), pp. 127–131.

[21] G. Schaefer and T. Nakashima, “Application of Cost-sensitive Fuzzy Classifiers to Image Understanding Problems,” Proc. Of IEEE International conference on Fuzzy Systems (FUZZ-IEEE 2009), Oct. 2009, pp. 1364–1368.

[22] T. Nakashima, G. Schaefer, Y. Yokota, and H. Ishibuchi, “A weighted fuzzy classifier and its application to image processing tasks,” Fuzzy Sets and Systems, vol. 158, Feb. 2007, pp. 284-294.

[23] O. Cordon, M. J. del Jesus, and F. Herrera, “A proposal on reasoning methods in fuzzy rule-based classification systems,” International Journal of Approximate Reasoning, vol. 20, Jan. 1999, pp. 21-45.

[24] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 94), Dec. 1994, pp. 487-499.

[25] H. Ishibuchi and T. Yamamoto, “Comparison of heuristic criteria for fuzzy rule selection in classification problems,” Fuzzy Optimization and Decision Making, vol. 3, June 2004, pp. 119-139.

[26] M. Zolghadri Jahromi and M. Taheri, “A Proposed Method for learning rule weights in Fuzzy Rule Based Classification Systems,” Fuzzy Sets and Systems, vol. 159, Feb, 2008. pp. 449-459