A MARGIN-BASED MODEL WITH A FAST LOCAL SEARCH FOR RULE WEIGHTING AND REDUCTION IN FUZZY RULE-BASED CLASSIFICATION SYSTEMS

M. TAHERI AND M. ZOLGHADRI JAHROMI

Abstract. Fuzzy Rule-Based Classification Systems (FRBCS) are highly investigated by researchers due to their noise-stability and interpretability. Unfortunately, generating a rule-base which is sufficiently both accurate and interpretable, is a hard process. Rule weighting is one of the approaches to improve the accuracy of a pre-generated rule-base without modifying the original rules. Most of the proposed methods by now, may over-fit on training data due to generating complex decision boundaries. In this paper, a margin-based optimization model is proposed to improve the performance on unseen data. By this model, fixed-size margins are defined along the decision boundaries and the rule weights are adjusted such that the marginal space would be empty of training instances as much as possible. This model is proposed to support the single-winner reasoning method with a special cost-function to remove undesired effects of noisy instances. The model is proposed to be solved by a fast well-known local search method. With this solving method, a huge amount of irrelevant and redundant rules are removed as a side effect.Two artificial and 16 real world datasets from UCI repository are used to show that the proposed method significantly outperforms other methods with proper choice of the margin size, which is the single parameter of this method.

1. Introduction

Although a large number of methods are proposed in the past for the task of classification [9, 23], proposing methods to construct interpretable classifiers is still a challenging task [1]. In a Fuzzy Rule-Based Classification System (FRBCS), the task is performed by a set of fuzzy rules [16, 5]. For the sake of interpretability, the classifier is usually designed by a small number of short (i.e., a small number of antecedent conditions) fuzzy rules. Using fuzzy sets having linguistic meaning must be used to make the classifier understandable. One main advantage of these systems is that prior knowledge about the problem in hand can be used to insert/modify fuzzy rules [24, 18].

It is a challenging task to design a fuzzy classifier having high classification accuracy when restricted to use a small set of short fuzzy rules having linguistic interpretably. This is why; many investigations have been conducted to extract fuzzy rules from

Received: July 2013; Revised: March 2014; Accepted: May 2014

Key words and phrases: Classification, Fuzzy systems, Rule weighting, Soft margin, Noise-stable, Local search.

more powerful classifiers such as Neural Networks [10, 17] and Support Vector Machine (SVM) [33].

To improve accuracy of FRBCSs, one can use a learning technique to adjust antecedent membership functions. However, this degrades interpretability of the resulting classifier. Rule weighting has often been used as an alternative to tune a FRBCs and preserve the interpretability to some extent. Assuming that an initial rule-base is available for the problem in hand, the task of rule weighting algorithm is to improve the performance of the initial rule-base by assigning a weight to each fuzzy rule [34, 15, 25] without modifying the initial rules. The rule weighting algorithms may remove redundant rules (by setting their weights to zero) to improve the efficiency and interpretability of the initial rule-base. Tuning the weights of the fuzzy rules can gather both of interpretability and performance in a classification system [15, 21].

A number of rule weight specification methods have been proposed in the past. Ishibuchi and Yamamoto [21] proposed four heuristic methods of rule weight specifications. These methods are fast but experiments show that they cannot significantly improve the classification accuracy.

Nozaki et al. [26] proposed a reinforcement learning approach for tuning the rule weights. But in their proposed method, hundreds of passes over training data are required to achieve an acceptable solution. It is impractical to use this method on large data sets. Also, the final solution depends on the order of training data presented to the algorithm. The algorithm is very sensitive to noisy (i.e., mislabelled) training data.

Genetic Algorithm (GA), as a search strategy suitable for complex search spaces, is widely investigated for constructing the rule-base in FRBCSs [20, 14, 22] and rule weighting [3, 4, 19]. Also a rule weighting algorithm based on gradient descent is proposed by one of the authors in [8], which attempts to minimize an objective function closely related to error-rate of the classifier on training data.

The authors have proposed a rule weighting algorithm in [34], which is based on a proposed novel greedy local search algorithm to minimize the classification error-rate on training data . In this paper, we denote this algorithm as Iterative Greedy Accuracy-based Rule-Weighting (IGARW). This algorithm achieves two desirable goals at the same time. Firstly, the classification rate is improved by adjusting the rule-weights. Secondly, redundant rules are removed during the learning process resulting in a compact rule-base. The novel optimization technique proposed in IGARW can be viewed as a general optimization scheme for problems having a discrete cost function.

The optimization technique originally proposed in IGARW is quite powerful in reducing the error-rate on training data. However, it fails to produce good generalization especially when dealing with problems having highly overlapped classes or noisy training data. This paper attempts to improve the generalization ability of the IGARW. This is achieved by considering a margin around the decision boundary and inserting it directly in the objective function.

Recently, intensive efforts are dedicated to improve the generalization of the classifiers with a margin-based optimization model [13]. A margin-based rule weighting

algorithm has been proposed by one of the authors in [27], which attempt to maximize a quadratic objective function inspired by SVM. In the rest of this paper, we denote this method as Fuzzy Kernel SVM (FKSVM).
FKSVM is based on weighted vote reasoning method. The method cannot be used for single winner reasoning commonly used in FRBCSs and believed to be more interpretable than weighted vote reasoning. The rule weighting scheme proposed in this paper is designed to use single-winner reasoning by sacrificing the convexity. FKSVM is based on a convex cost function for the instances violating the margin. A noisy instance may add a large cost to the objective function and greatly affect the decision boundary. Although, alternative cost functions are proposed in the literature to tackle the negative effect of outliers [32], they violate the convexity of the objective function. In other words, these objective functions cannot be used with FKSVM.
The solution we provide in this paper is based on a cost function that is resistant to noisy data. As the model is no longer convex, a modified version of IGARW is proposed (with the same time complexity as IGARW) for rule weighting and removing redundant rules.
This paper is organized as follows. In section 2, the structure of fuzzy rule-based classifiers is briefly introduced. In section 3, the basic concepts of IGARW are provided. In section 4, SVM and FKSVM are also briefly described to introduce the basic notation of margin-based optimization. In section 5, our proposed method is presented in detail. The results of experiments are reported in section 6. Finally, the conclusion is given in section 7.

2. **Fuzzy Rule-based Classification Systems (FRBCS)**

Fuzzy rule-based classifiers are composed of three main parts: database, rule base and reasoning method. The database contains a series of fuzzy sets associated with specific linguistic terms (e. g. small, medium, large). The rule base is a collection of fuzzy classification rules used along with reasoning method to classify input data.
Various rule types are proposed in the literature for classification [16, 34, 15]. The structure of fuzzy rules we use in this paper is as follows:

$$Ruler_k: \ If\ x^1 = A_k^1\ \&\ x^2 = A_k^2\ \&\ \ldots\ \&\ x^d = A_k^d \Rightarrow class\ h_k\ with\ (\omega_k) \quad (1)$$

Where, r_k denotes k^{th} rule in the rule-base, x^i is the i^{th} feature value of the input pattern $X = \left[x^1, x^2, , x^d\right]$, d is the number of features, A_k^i is the fuzzy set used for the i^{th} feature in the rule r_k, h_k is a class label as the consequence part of the rule and ω_k denotes weight of the rule. In order to specify how much an instance X is compatible with a rule, a T-Norm function is commonly used. The T-norm function used in this paper is the algebraic product. Using this, the compatibility grade of input pattern X with the rule r_k can be shown as:

$$\mu_k(X) = \prod_{1 \le i \le d} \eta\left(x^i, A_k^i\right) \quad (2)$$

Where $\mu_k(X)$ denotes the compatibility grade of instance X with r_k and $\eta\left(x^i, A_k^i\right)$ denotes the degree of the membership of x^i in the fuzzy set A_k^i. Any other T-norm

can be used here regardless of the rule weighting method. The weight of the rule is used to modify the compatibility grade (i.e. called as weighted compatibility grade) by multiplying the compatibility grade with the rule weight (i.e. $\omega_k \mu_k(X)$).
The reasoning method is the strategy of classifying an instance based on weighted compatibility grades. There are two major types of reasoning [21]: single-winner and weighted-vote. In the single-winner method, each instance is classified as the consequent class of the winner rule (which has the maximum value of weighted compatibility grade). This can be formally stated as:

$$X \text{ is classified as } h_{k^*} \text{ where } k^* = \underset{k}{argmax}\left[\omega_k \mu_k(X)\right] \tag{3}$$

In the weighted vote method, all the rules vote for classifying the input data. The vote for each class is the sum of the weighted compatibility grades of the rules in that class. This can be formally stated as:

$$X \text{ is classified as } c^* \in CL \text{ where } c^* = \underset{c \in CL}{argmax}\Big[\sum_{r_k \in R^c} \omega_k \mu_k(X)\Big] \tag{4}$$

Where, $CL = \{c_1, c_2, \ldots, c_{|CL|}\}$ is the set of all class labels, and R^c is the set of the rules having class c in the consequent ($R^c = \{r_k | h_k = c\}$).

3. Iterative Greedy Accuracy-based Rule Weighting (IGARW)

Assuming that an initial rule-base for the problem in hand is available, IGARW attempts to minimize the error-rate of the classifier on training data by specifying the weights of the rules in the rule-base. Finding the optimal set of the rule-weights is not easy. IGARW uses a greedy descent optimization approach. In descent optimization methods, an initial solution is improved iteratively such that in each iteration the objective value does not increase. In IGARW, only one of the rule-weights, in each step, is set to its optimal value considering other weights fixed. Assume that the rule-weights $[\omega_1, \omega_2, \ldots, \omega_{|R|}]$ are the only parameters which should be adjusted. As the initial solution, all the rule-weights are set to 1.In the first step, ω_1 is set to its optimal value assuming other weights are fixed. The optimal value is the one, which minimizes the error on the training data. In the next step, ω_2 is set to its optimal value considering other rule weights (even ω_1 with its new value) are fixed. This process is done for all the weights in $|R|$ steps. It is possible that, optimality of ω_1 and some of other rule weights will not be preserved at the final step. Therefore, the process of weighting can be repeated again in more iterations. Since the objective value does not increase during the optimization, the set of rule-weights converges to a local optimum solution in a few number of iterations. Finding the global solution is not desired, because even with a local optimum one, the system may over-fit on training data. The novelty of IGARW is the method of finding the optimal value of each rule-weight considering others fixed. In the following, this method is explained from a new point of view that is beneficial to solve the proposed model.

3.1. Computing the Optimal Value of a Rule Weight.

Assume that all the rule-weights except that of ω_k are given fixed and it is desired to set ω_k to its optimal value in the range $[0, +\infty]$. The objective function is the classification error on the training instances. Error of each instance is 1 if and only if it is misclassified as shown in (5).

For each X_j, the rules are partitioned into two sets: friend and enemy rules. A friend rule is a rule with the consequence label equal to y_j (i.e. the class of X_j). Other rules are called enemies of X_j. Maximum weighted compatibility grades of X_j with friend and enemy rules are used to produce a criterion (ϵ_j) to measure the distance between X_j and the decision boundary. Positive and negative values of this distance indicate misclassification or correctly classification of X_j, respectively. This model is presented in (5).

$$\begin{aligned} &minimize \quad E = \sum_{X_j} \ell\left(\epsilon_j\right) \\ &s.t.\ \forall X_j: \quad \epsilon_j = \underset{r_i \notin R^{y_j}}{max}\left(\omega_i \mu_i\left(X_j\right)\right) - \underset{r_i \in R^{y_j}}{max}\left(\omega_i \mu_i\left(X_j\right)\right) \end{aligned}$$

$$\ell\left(\epsilon_j\right) = \begin{cases} 0 & if \quad \epsilon_j < 0 \\ 1 & if \quad \epsilon_j \geq 0 \end{cases} \tag{5}$$

where, E is the classification error on the training data as the sum of the classification costs of the training instances. Cost of each training instance is 0 or +1 if it is correctly classified or not, respectively. The step function $\ell\left(\epsilon_j\right)$ is a cost function, which maps the distance between X_j and the decision boundary to a classification cost.

To find the optimal value of ω_k, its value increases from 0 to $+\infty$ and the classification cost on each training instance is traced. If X is not covered by the rule r_k (i.e. $\mu_k\left(X\right) = 0$), the value of ω_k does not effect on the classification result of X and it is ignored here. Otherwise, by increasing the value of ω_k, the weighted compatibility grade of X with the rule r_k increases until r_k becomes the strongest enemy or friend of X at $\omega_k = th^{cut}$. If r_k is a friend rule of X_j, this threshold is computed by (6).

$$th^{cut} = \frac{\underset{r_{i \neq k} \in R^{y_j}}{max}\left[\omega_i \mu_i\left(X_j\right)\right]}{\mu_k\left(X_j\right)} \tag{6}$$

For $\omega_k < th^{cut}$, the distance between X_j and the decision boundary does not change. However, for values of ω_k in the range $(th^{cut}, +\infty)$, the distance may vary such that the classification cost may be toggled at $\omega_k = th^{cut}$ as shown in (7).

$$th = \frac{\underset{r_{i \neq k} \notin R^{y_j}}{max}\left[\omega_i \mu_i\left(X_j\right)\right]}{\mu_k\left(X_j\right)} \tag{7}$$

If th is less than th^{cut}, increasing ω_k has no effect on the cost of X_j. Otherwise, the cost of X_j differs for some value less than or greater than th. In contrary with the previous case, if r_k is an enemy rule of X_j, the thresholds th^{cut} and th are computed by (7) and (6), respectively.

If r_k is a friend or enemy rule of X_j, the objective value decreases or increases, respectively, at $\omega_k = th$. This is why; a cost -1 or +1 is attached to the threshold th if r_k is a friend or enemy rule, respectively, as shown in Figure 1. Due to that, some thresholds may be equal or very near to each other, some intervals may be empty or very small. The optimal interval is selected from the ones, which are larger than a predefined size.

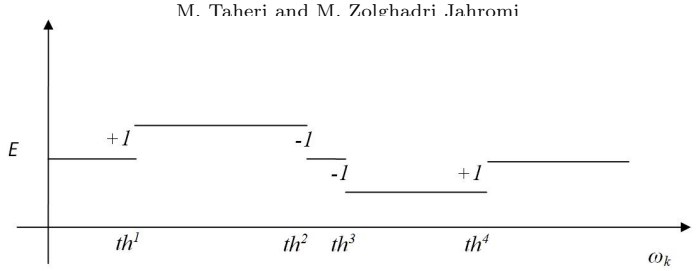


FIGURE 1. Sample of Classification Error (E) on the Training Data vs. ω_k

Setting ω_k to any value from the optimal interval optimizes the objective function respect to ω_k. In IGARW, if the optimal interval is between two consecutive thresholds $\left(th^j, th^{j+1}\right)$, ω_k is set to the mean of its values i.e. $\left(th^j + th^{j+1}\right)/2$. If the intervals $\left[0, th^1\right)$ or $(th^n, +\infty)$ are optimal, ω_k will be set to 0 or $th^n + \tau$, respectively (where, τ is a large number and th^n is the greatest threshold). Setting some weights to zero is equal to removing associated rules from the rule base as a side-effect of this rule weighting method.

4. **Support Vector Machine and Fuzzy Kernel SVM**

In this paper, inspiring by large margin classification systems such as SVM [29, 6], marginal based nearest neighbour [30] and Fuzzy kernel SVM [27], a margin is designed along the decision boundaries. It is desired to separate the instances from different class labels such that instances are located out of the margin as much as possible. In these learning methods, the margin is maximized. A general model of the large margin classifiers can be explained as shown in (8).

$$\begin{array}{ll} Maximize & M \\ s.t. \quad \forall X_j, c \neq y_j : & f_{c,y_j}(X_j) \geq M \end{array} \tag{8}$$

where, $f_{c,y}(X)$ presents the distance between X and the decision boundary, which separates the class labels c and y. For example, in binary linear SVM, this decision boundary is a hyper-plane, which separates the instances of two class labels +1 and -1 as shown in (9) and figured in Figure 2.

$$\begin{array}{ll} Maximize & M \\ s.t. \quad \forall X_j : & \dfrac{y_j\left(W^T X_j + b\right)}{\|W\|} \geq M \\ & y_j \in \{+1, -1\} \end{array} \tag{9}$$

where, W and b are the perpendicular vector and bias of the discriminator hyper plane. For any hyper-plane, W and b can be scaled without any change in the decision boundary. Without loss of generality, it is assumed that W of the optimal hyper plane is such that $\|W\| M = 1$. With this constraint, the problem (9) is reformulated as (10).

$$\begin{array}{ll} Maximize & \dfrac{1}{\|W\|} \\ or & \\ Minimize & \frac{1}{2}W^2 \\ s.t. \quad \forall X_j : & y_j\left(W^T X_j + b\right) \geq 1 \\ & y_j \in \{+1, -1\} \end{array} \tag{10}$$

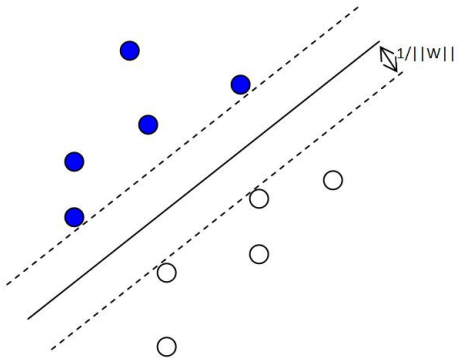


FIGURE 2. SVM Classifier with the Maximum Margin for Linear Separable Data Points

Inspiring by Multi-class SVM [12] specially the one proposed by Weston et al. [31], a margin-based rule weighting in fuzzy classifiers is proposed in [27] for only weighted-vote reasoning method. In that paper, $f_{c,y}(X)$ is defined as the normalized differences of the strengths (accumulated weighted compatibility grades) of the class labels c and y on the instance X as shown in (11).

$$\begin{aligned} & Maximize && M \\ s.t. \quad & \forall X_j, c \neq y_j : && \frac{\sum\limits_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) - \sum\limits_{r_i \in R^{c}} \omega_i \mu_i(X_j)}{\|\Psi\|} \geq M \\ & && \omega_i \geq 0 \end{aligned} \tag{11}$$

where, $\|\Psi\|$ is the norm of the vector $\Psi = [\omega_1, \omega_2, \ldots, \omega_{|R|}]$ as the vector of the rule-weights. In following, this model is adapted for single-winner reasoning.

5. The Proposed Rule Weighting Method

The proposed method is a margin-based optimization model inspired by FKSVM that is modified to support the single-winner reasoning method. Since, this model is not convex, it is solved by the greedy descent method proposed in IGARW. It should be emphasized that, this solver is adapted to solve this new objective function.

5.1. **The Proposed Model.** In weighted-vote reasoning, the set of the rules belonged to two different class labels compete against each other to be the winner. However, in single-winner reasoning, a single rule is the winner component, which determines the class label of the instance. This is why; decision boundaries in single-winner are defined based on the maximum compatibility grades between friend and enemy rules as shown in (12). Unfortunately, this model could not be convex.

$$\begin{aligned} & Maximize && M \\ s.t. \quad & \forall X_j : && \frac{\max\limits_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) - \max\limits_{r_i \notin R^{y_j}} \omega_i \mu_i(X_j)}{\|\Psi\|} \geq M \\ & && \omega_i \geq 0 \end{aligned} \tag{12}$$

Due to have non-negativity constraints on the rule-weights($\omega_i \geq 0$), the first norm of Ψ is equal to the sum of weights. With this norm, the constraints are linear for the constant value of M. This is why; the first norm is preferred in this paper to the second norm that is used in both SVM and FKSVM.
In large margin models, the constraints are relaxed by an error for each instance, which cannot completely satisfy associated constraint. There is a regularization parameter in (FK)SVM to determine a trade-off between errors of instances and the margin-size. In this paper, instead of regularization, the margin-size M is considered as the single parameter and the objective function is defined as minimizing the errors of instances e_j. The final model is formulated in (13).

$$\begin{aligned} &Minimize && \sum_{X_j} e_j \\ &s.t. \quad \forall X_j : && \frac{\max\limits_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) - \max\limits_{r_i \notin R^{y_j}} \omega_i \mu_i(X_j)}{\sum \omega_i} \geq M - e_j \\ & && \omega_i \geq 0 \quad e_j \geq 0 \end{aligned} \tag{13}$$

With a modification, this problem can be reformulated based on a cost function in (14).

$$\begin{aligned} &Minimize && \sum_{X_j} \Gamma(\epsilon_j) \\ &s.t. \quad \forall X_j : && \epsilon_j = \frac{\max\limits_{r_i \notin R^{y_j}} \omega_i \mu_i(X_j) - \max\limits_{r_i \in R^{y_j}} \omega_i \mu_i(X_j)}{\sum \omega_i} \\ & && \omega_i \geq 0 \\ &where && \Gamma(\epsilon) = \begin{cases} 0 & if \quad \epsilon < -M \\ \epsilon + M & if \quad \epsilon \geq -M \end{cases} \end{aligned} \tag{14}$$

This model is similar to (5). Each training instance has a relative error ϵ_j that produces some cost in the objective function using the non-decreasing cost function Γ. This cost function is the same as the one used in SVM and FKSVM and is depicted in Figure 3. With this cost function, an outlier instance which is misclassified with a large value of related error, has a great effect on the optimal solution. Without any change in the model, the cost function is such scaled that $\Gamma(-M) = 0$ and $\Gamma(M) = 1$ as depicted in Figure 3. Inspiring by [32], a piecewise linear cost function is presented in (15) to ignore the extra effects of noisy and outlier instances.

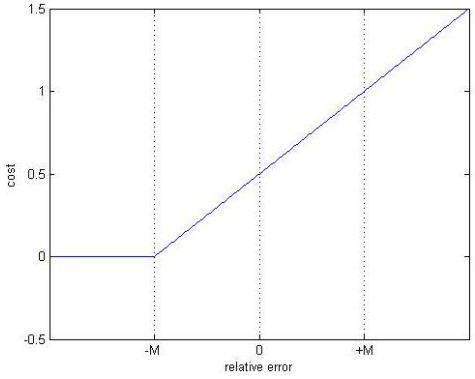


FIGURE 3. The Cost Function Used in SVM and FKSVM

$$\Gamma^{FL}(\epsilon) = \begin{cases} 0 & if \quad \epsilon < -M \\ (\epsilon + M)/2M & if \quad -M \leq \epsilon \leq M \\ 1 & if \quad M < \epsilon \end{cases} \tag{15}$$

This cost function is a linear function, which is filtered from both sides for values less than M and greater than M. This is why; this cost function is called Filtered Linear (FL) function, in this paper. With this approach, no instance can increase the objective value more than one unit even if it has a large relative error. Unfortunately, this cost function is not convex.

Both of supporting single-winner reasoning and use of FL cost function, makes the proposed model a non-convex optimization problem. Consequently, it may have more than one local optimum solutions. In the following, at first the non-convex objective function is approximated by a discrete stepwise function. Then the greedy descent method described in section 3, is adapted to solve this model.

5.2. **Discrete Approximation.** In following, FL cost function is approximated by averaging on $2p + 1$ step functions. This function is called here p-approximation, as presented in (16) and depicted in Figure 4.

$$\widetilde{\Gamma}^{p}(\epsilon) = \frac{1}{2p+1} \sum_{t=-p}^{p} \ell\left(\epsilon - t\frac{M}{p}\right) \tag{16}$$

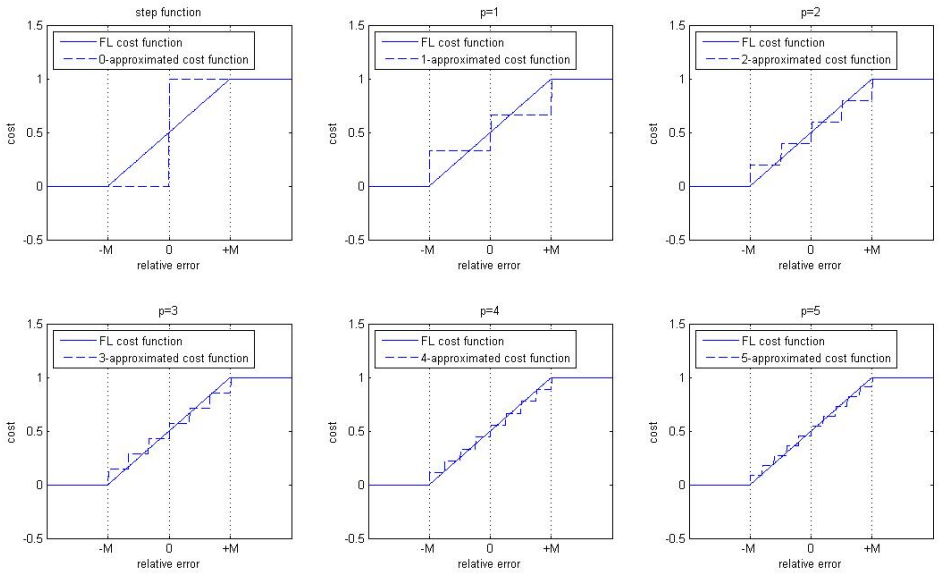


FIGURE 4. The Cost of a Training Instance vs. Its Relative Error Based on FL and p-approximation Cost Functions

As shown in Figure 4, FL cost function can be well approximated by sufficiently large values of p. After this approximation, the solver used in IGARW can be applied to this model without extra time complexity. It should be underlined that, the proposed objective function is the same as the one in IGARW for $M = 0$ (having no margin) or 0-approximation (i.e. one step function).

5.3. **Solving the Model.** The same as the strategy presented in IGARW, initial values of all the rule-weights are set to 1. Then the rules are considered one by one. In each step, one of the rule-weights is set to its optimal value considering other rule-weights are given and fixed. Optimal value of each rule weight ω_k optimizes the following model respect to only ω_k.

$$\begin{aligned} Minimize \quad & \sum_{X_j} \sum_{t=-p}^{p} \ell(\epsilon - t\overbrace{\frac{M}{p}}^{t'}) \\ s.t. \quad \forall X_j : \quad & \epsilon_j = \frac{\max\limits_{r_i \notin R^{y_j}} \omega_i \mu_i(X_j) - \max\limits_{r_i \in R^{y_j}} \omega_i \mu_i(X_j)}{\sum \omega_i} \\ & \omega_i \geq 0 \end{aligned} \tag{17}$$

By increasing ω_k, the weighted compatibility grade of X_j with r_k increases. There are some threshold values th whereas for ω_k less than or greater than th, one of the step functions is toggled and consequently the objective value increases or decreases by one. Based on these threshold, the range of values of ω_k is divided into some intervals, in which the objective function is fixed. It is desired to find the optimal interval for values of ω_k. If ω_k is set to any value of this interval, the sum of the step functions presented in (17) is minimized.

For each step function separately, there are some thresholds on values of ω_k. A cost +1 or -1 is attached to a threshold if passing ω_k from that threshold, the objective function increases or decreases, respectively. Each step function is identified by associated training instance X and the value of $t' = t\frac{M}{p}$ in (17). Thresholds of such step function can be computed as follows.

Let us assume that r_k is a friend rule of X ($h_k = y$). The same as IGARW, there is a threshold th^{cut} that, for $\omega_k < th^{cut}$, the rule r_k is not the strongest friend rule (with the maximum weighted compatibility grade). This threshold is similarly computed by (6).

However, in contrary with IGARW, changing ω_k in the range $[0, th^{cut})$ may toggle the values of some step functions due to the normalization factor $\sum \omega_k$. These critical thresholds are computed by (18).

$$th_1 = \frac{\max\limits_{r_{i\neq k} \notin R^y} \omega_i \mu_i(X) - \max\limits_{r_{i\neq k} \in R^y} \omega_i \mu_i(X)}{t'} - \sum_{i \neq k} \omega_i \tag{18}$$

The thresholds computed by (18), which are smaller than th^{cut} are acceptable. For ω_k greater than th^{cut}, different types of thresholds are computed using (19).

$$th_2 = \frac{\max\limits_{r_{i\neq k} \notin R^y} (\omega_i \mu_i(X)) - t' \sum\limits_{i \neq k} \omega_i}{t' + \mu_k(X)} \tag{19}$$

This threshold is acceptable if it is greater than th^{cut}. With the similar reasoning, if r_k is an enemy rule of X, both of thresholds th^{cut} and th_1 are computed as mentioned above. However, th_2 is computed by (20).

$$th_2 = \frac{\max\limits_{r_{i\neq k} \in R^y} (\omega_i \mu_i(X)) + t' \sum\limits_{i \neq k} \omega_i}{-t' + \mu_k(X)} \tag{20}$$

Acceptable thresholds th_1 and th_2 for each step function are stored in a list and the optimal value of ω_k is found the same as IGARW. The pseudo-code of this learning algorithm is presented in Table 1. As shown in the pseudo-code, in I iterations, each rule r_k is weighted. For each rule, $O(|T|.p)$ thresholds are computed by the nested

loops 10 and 11. For each threshold, the friend and enemy rules with the maximum compatibility grades on associated instance should be found. This process can easily be done in $O(|R|)$. Hence, the total time-complexity of finding the thresholds is $O(I.|T|.p.R^2)$. Using a heap structure for each training instance to store the rule indexes based on their weighted compatibility grades, this time-complexity reduces to $O(I.R.|T|.(p + lgR))$. The complexity of the rest of the code can be ignored.
It can be concluded that the time-complexity of the proposed method (i.e. the same as the one of IGARW), is linearly dependent on the size of the training data and also approximately linearly dependent on the size of the initial rule-base. This is why; this method is highly efficient and scalable in comparison with other methods such as Reward & Punishment (R&P)[26], which should be run in hundreds of iterations to achieve a rational convergence with a good solution.

6. **Experimental Results**

In this paper, the proposed method of rule weighting is compared with the initial rule base with equal rule-weights (before weighting), 4 greedy methods of rule weighting [21] called as G1-G4, reinforcement learning method R&P [26], and IGARW [34].
Two artificial datasets and 16 standard datasets of UCI repository [28] as shown in Table 2 are used to compare interpretability, learning power, generalization on test data and stability of the proposed method with others.

6.1. **Rule Generation.** There are many rule construction methods e.g. based on GA [2]. In the rule weighting method, it is usually assumed that the rule-base is constructed and given. In this paper, the rule-base is generated by an efficient method [19, 11, 3], which is used from 1992. By this method, the rule-base has a high degree of interpretability, because each rule has at most two antecedents. Totally, 14 fuzzy sets are considered for each dimension as shown in Fig 5.

All rules with at most two antecedents are initially generated. The consequence of each rule is the class label with the maximum confidence on the training data. Then, the rules are sorted based on the confidence in descending order and the first 30 rules from each class label are selected.

6.2. **Interpretability and Learning Power.** Learning power usually indicates the power of the final model to classify the training data. As described in Table 3, IGARW have proper classification rate on the training data. However, the real learning power is measured by the final value of the objective function. There is no explicit objective function for the methods G1-G4 and R&P. In IGARW, the accuracy of the model on the training data is directly optimized. The proposed method MIGRW sacrifices the classification of the training data in order to achieve a relative clean margin to increase the generalization.
In addition, the size of rule-base is used in this paper as a criterion for measuring interpretability. The smaller size of the final rule-base, the higher interpretability is achieved. The methods, G1-G4 and R&P methods do not reduce the size of the rule-base. In Table 3, the classification rates on the training data and the sizes of the final rule-bases are compared after learning using all of the instances in the dataset.

function RULE_WEIGHTING(R,T,M,p)
▷ The set of fuzzy rules $R = \{r_1, r_2, \ldots, r_{|R|}\}$
▷ The training set $T = \{X_1, X_2, \ldots, X_{|T|}\}$
▷ M as the margin-size and p as the approximation parameter
▷ Output is the set of the rule-weights $\Psi = \{\omega_1, \omega_2, \ldots, \omega_{|R|}\}$
$\forall r_k : \omega_k \leftarrow 1$ ▷ Initialize all the rule weights to 1
for I iterations **do**
for each $r_k \in R$ **do**
Ł $\leftarrow$ an empty set
for each $X \in T$ **do**
for $t \leftarrow -p$ *to* p **do**
$t' \leftarrow t\frac{M}{p}$
compute th^{cut}, th_1 and associated $cost_1$
if $0 \leq th_1 < th^{cut}$ **then**
insert $(th_1, cost_1)$ to L
end if
compute th^{cut}, th_2 and associated $cost_2$
if $th_2 > th^{cut}$ **then**
insert $(th_2, cost_2)$ to L
end if
end for
end for
Sort L in ascending order respect to threshold values.
▷ $L = [< th^i, cost^i > | 1 \leq i \leq n]$ where $i < j \rightarrow th^i \leq th^j$
Find the best interval in L
if best interval=$[0, th^1)$ **then**
$\omega_k \leftarrow 0$
else if best interval=$(th^n, +\infty)$ **then**
$\omega_k \leftarrow th^n + \tau$
else if best interval=(th^i, th^{i+1})) **then**
$\omega_k \leftarrow \frac{th^i + th^{i+1}}{2}$
end if
end for
end for
return Ψ

end function

TABLE 1. Pseudo-code of Rule Reduction

Initial rule-base contains 30 rules per class. In the following, results of MIGRW are achieved with the margin size $M = 0.063$ and 1-approxiamtion. In addition, 3 iterations of rule weighting are done in both of IGARW and MIGRW. Also, the number of iterations for R&P method has been set to 500. Because the methods G2, G3 and G4 are similar for the datasets with two class labels, the redundant results are not reported.

Dataset	Pattern No.	Feature No.	Class No.	Dataset	Pattern No.	Feature No.	Class No.
Haberman	306	3	2	Iris	150	4	3
Pima	768	8	2	Wine	178	13	3
Bcancer	684	10	2	Lung Cancer	32	56	3
WDBC5	569	30	2				
WPBC	198	33	2	Heart	287	13	5
Ionosphere	351	34	2	Glass	214	9	6
Sonar	208	60	2	Image	210	19	7

TABLE 2. List of UCI Datasets Used in This Paper

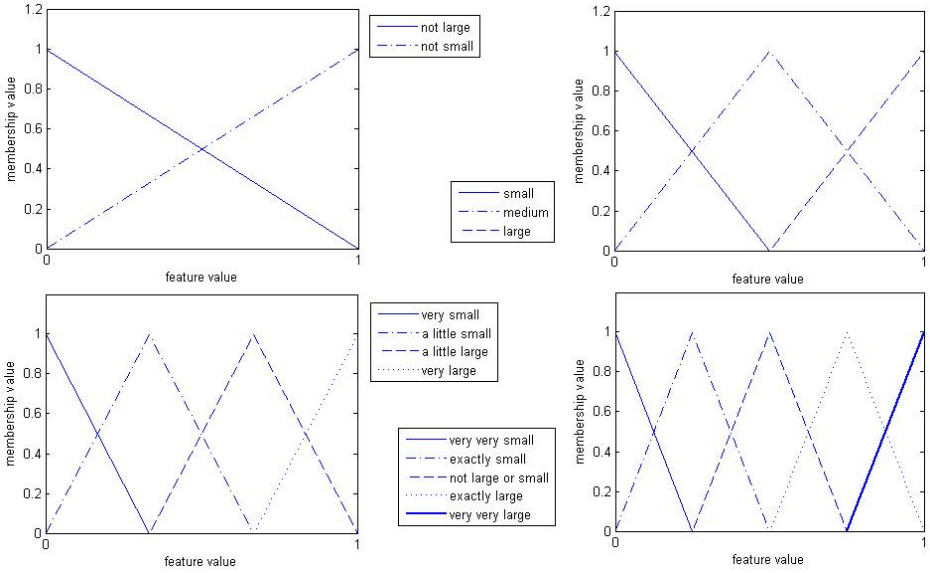


FIGURE 5. Fourteen Fuzzy Sets Used in This Paper for Rule Generation

As shown in Table 3, IGARW outperforms other rule weighting methods in classification on the training data with a small set of the rules. Although MIGRW has a lower classification rate in this table, it is shown, in the next section, that it has a better performance on unseen data if a good margin-size is found. Both of the methods IGARW and MIGRW, generate a compact rule-base in comparison with the initial one. For example, MIGRW removes all the rules except of only one rule per class for the dataset Iris, as presented in Table 4.

Accuracy of MIGRW is not as well as IGARW on the training data. However, it should be underlined that the objective function of MIGRW is the one presented in (17) that is called here as margin-based error. In Figure 6, classification errors and margin-based errors of both of IGARW and MIGRW on Glass dataset are traced during the learning process.

Dataset	No Weight		G1	G2	G3	G4	R&P	IGARW		MIGRW	
	Rate	Rule#	Rate	Rate	Rate	Rate	Rate	Rate	Rule#	Rate	Rule#
Haberman	73.53	60	74.18	74.18	-	-	77.45	80.72	9	73.86	60
Pima	65.89	60	67.06	69.66	-	-	72.40	79.04	9	75.00	55
BCancer	94.01	60	94.30	94.30	-	-	96.35	98.10	6	96.93	4
WDBC5	93.15	60	93.67	93.32	-	-	93.85	96.84	6	94.02	13
WPBC	77.27	60	77.27	77.27	-	-	82.32	86.36	9	84.34	8
Ionosphere	67.52	60	86.61	87.18	-	-	88.03	89.46	6	87.75	25
Sonar	70.67	60	75.48	77.40	-	-	65.87	83.17	8	77.88	22
Parkinson	75.38	60	75.38	75.38	-	-	83.59	89.23	4	87.69	8
Promoters	68.87	60	72.64	73.58	-	-	92.45	97.17	8	96.23	8
Spect(Float)	79.40	60	79.40	79.40	-	-	83.52	91.01	14	80.15	7
Iris	94.67	90	95.33	96	95.33	96	99.33	98.67	5	96.67	3
Wine	94.94	90	96.07	95.51	95.51	96.07	100	100	8	97.75	9
Lung Cancer	53.13	90	96.88	93.75	96.88	96.88	84.38	100	9	96.88	7
Heart	57.14	150	60.28	61.67	60.63	61.67	58.89	66.90	16	55.75	139
Glass	62.61	180	62.61	65.42	63.55	62.62	59.35	74.30	13	64.02	9
Image	77.62	210	79.52	79.05	80.00	78.10	81.43	81.43	8	79.52	11
Average	75.36	88.13	80.42	80.82	80.85	80.81	82.45	88.28	8.63	84.03	24.25

TABLE 3. Classification Rate on the Training Data After Rule Weighting by G1-4 [21], R&P [26], IGARW [34] and MIGRW

Rule # 2	If x^3 is very small then class 1 with confidence 92.64
Rule # 37	If x^3 is not small or large & x^0 is medium then class 2 with confidence 176.7
Rule # 61	If x^3 is large then class 3 with confidence 167.8

TABLE 4. Three Rules Extracted by MIGRW for Classifying Instances of Iris Dataset

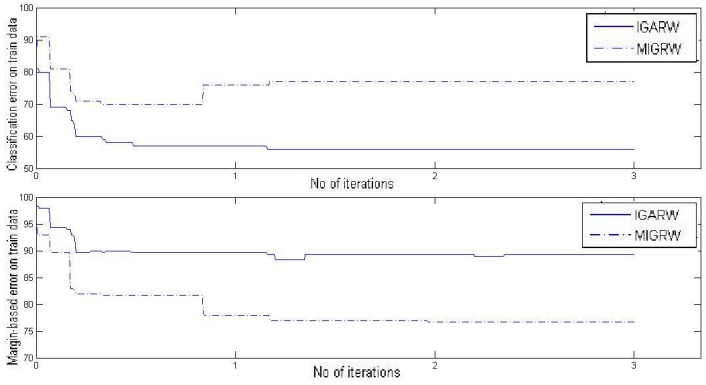


FIGURE 6. Trace of Classification and Margin-based Errors of IGARW and MIGRW on Glass Dataset vs. no. of Weighted Rules During Iterations

As depicted, each method outperforms the other one respect to its own objective function. Because both of them are descent optimization methods, associated objective values never increase during the learning process.

In order to emphasize the merits of MIGRW in comparison with IGARW, a 1-dimentional dataset is created as follows: $\{0.05, 0.1, 0.15, 0.85, 0.9, 0.95\}$ with the class label + and $\{0.4, 0.45, 0.5, 0.55, 0.6, 0.65\}$ with the class label $*$. An outlier instance $\{0.35\}$ with class label + is inserted to this dataset. Three one antecedent rules are generated as shown in Table 5. The decision space of each rule is determined in Figure 7, with different values of ω_2.

Rule # 1	If x is small then class + with confidence 1
Rule # 2	If x is medium then class $*$ with confidence ω_2
Rule # 3	If x is large then class + with confidence 1

TABLE 5. Three Rules Generated for the Artificial Dataset in Figure 7

As depicted, increasing the weight of the second rule, increases its decision space. The misclassified instances are marked by circles in Figure 7, which are $x = 0.35$ in (a) and $x = 0.65$ in (c). In Figure 7 (b), both of these instances are located on the decision boundary. Hence, 0.43 is a threshold for both of these instances. For ω_2 less than or greater than 0.43, one of these instances are misclassified. In weighting ω_2, there are two optimal interval (0.25,0.43) and (0.43,2.33) with only one misclassified instance. In IGARW, the smallest interval (0.25,0.43) is considered as the optimal and ω_2 is set to its mean value 0.34 as depicted in Figure 7(d). However, considering a margin-based error with a suitable margin-size, the interval (0.43,2.33) is preferred to (0.25,0.43). Considering the artificial dataset and the rule-base in Figure 7, the margin-based error for a set of margin-size values are depicted vs. ω_2 in Figure 8.

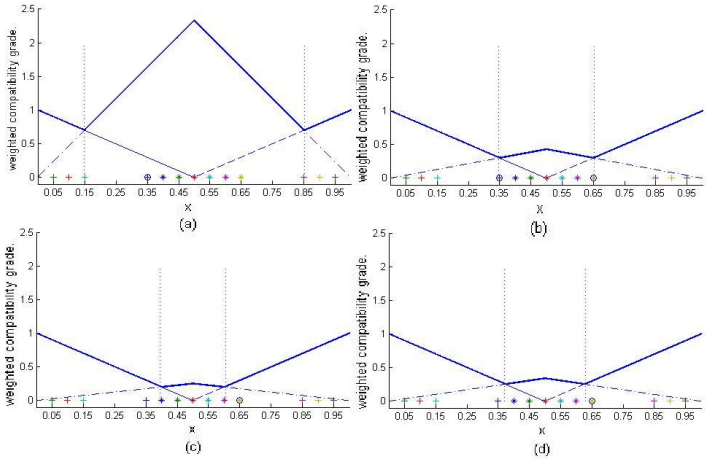


FIGURE 7. Decision Spaces of the Rules in Table 5 on a 1-dimensional Dataset with Different Weights of the 2^{nd} Rule: (a) $\omega_2 = 2.33$ (b) $\omega_2 = 0.43$ (c) $\omega_2 = 0.25$ (d) $\omega_2 = (0.25 + 0.43)/2 = 0.34$

The margin-based error with margin-size 0 is equal to the classification error. As shown in Figure 8, this error is optimal in the range (0.25,2.33). By increasing the

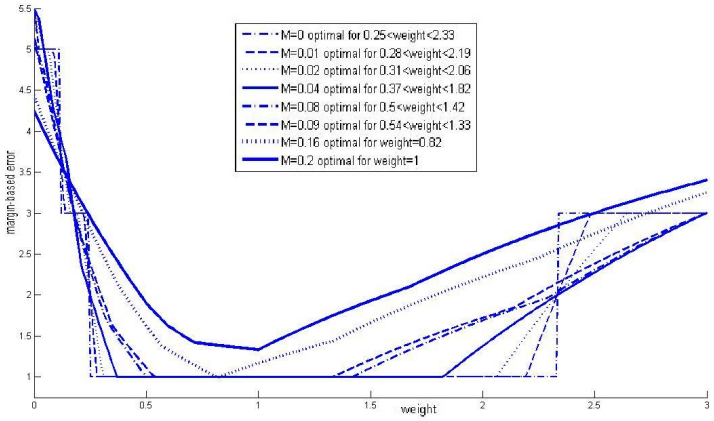


FIGURE 8. Margin-based Error vs. the Weight of the 2^{nd} Rule for the Artificial Dataset Presented in Figure 7

margin-size, the optimal interval converges to the point ω_2=0.82 at M=0.16. For larger values of M, the objective value of the optimal solution increases. Decision spaces of the rules on the above dataset are depicted in Figure 9, for ω_2=0.82. In this figure, a large margin around the decision boundaries are empty of the instances.

6.3. **Generalization.** In order to measure the generalization of the proposed rule weighting method and the others, two experiments have been provided:

(1) Comparison of the generated decision boundary with the optimal one on an artificial dataset.
(2) Comparison of classification rate in 10fold-10CV on 16 UCI datasets presented in Table 2.

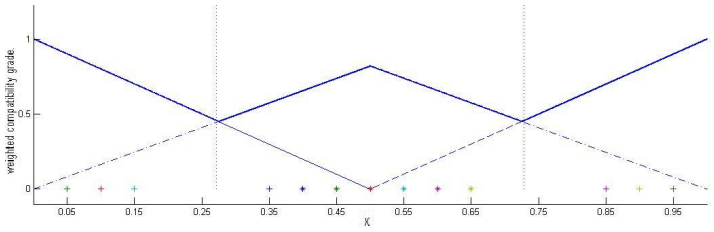


FIGURE 9. Decision Spaces of the Rules on the Artificial Dataset with ω_2=0.82

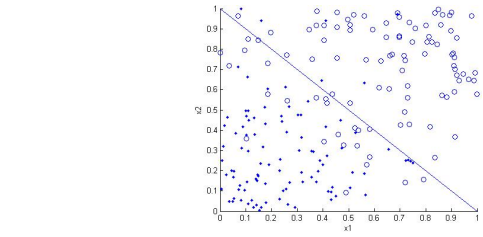
In the former experiment, an artificial 2-dimentional dataset is generated as follows. 100 **dot** and **circle** instances are randomly generated, with a normal distribution and similar variance, around centers (0,0) and (1,1), respectively. It is obvious that the optimal decision boundary is the line-segment connecting (0,1) to (1,0) as shown in Figure 10(a). Two rules are generated for this classification as presented in Table 6.

Rule #1	If x^1 is not large & x^2 is not large then class . with confidence 1
Rule #2	If x^1 is not small & x^2 is not small then class O with confidence ω_2

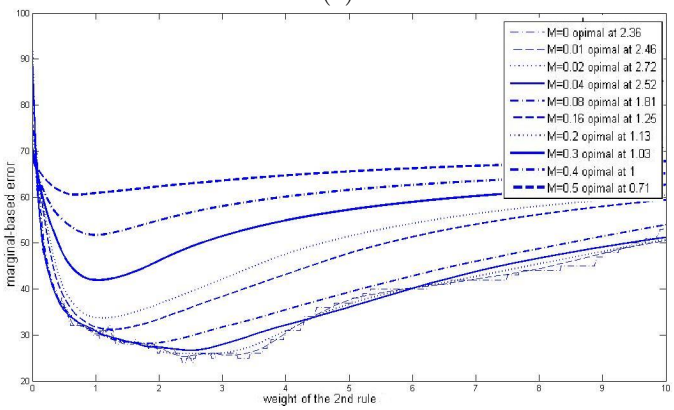
TABLE 6. Three Rules Generated for the Artificial Dataset in Figure 10

The generated decision boundary between the rules in Table 6 is optimal only for ω_2=1. However, as depicted in Figure 10(b), with IGARW (i.e. M=0) the optimal value of ω_2 is 2.36. By increasing the margin-size up to M=0.4, the optimum value of ω_2 approaches to 1.
Classification result of the proposed method with others on real datasets, are reported in Table 7 with Ten fold-Ten cross validation. IGARW is a special type of MIGRW with the margin-size equal to 0. However, selecting a proper margin-size can increase the generalization of the final rule-base on unseen data.
The margin-size is the only parameter of this method. Finding the best value of this parameter is as hard as finding the value of the regularization parameter in SVM. In this paper, the size of the margin is selected, by a cross validation on the training data, from the set of values 1/4, 1/8, 1/16 and 1/32. Hence, two nested cross validation is done to achieve the results reported in Table 7. The proposed method significantly (tested by t-test with confidence level 0.95) outperforms other methods on all used datasets. This improvement is represented by + sign in Table 7. The value of p in approximation is set to 1, 2 or 4 such that the difference of two consecutive breakpoints would be at most 1/16$\simeq$ 0.063. Hence, p is set to 1 for M=1/32 and 1/16. Also, p is set to 2 and 4 for margins M=1/8 and 1/4, respectively. As shown, the methods R&P and IGARW result in the same classification-rate on these datasets. However, R&P is highly time-consuming in comparison with IGARW. Paired T-test can be used in order to statistically comparing two methods over only one dataset. It is not recommended in recent researches [7], [11] for comparing classifiers on multiple datasets. In this paper, two non-parametric methods are used, to compare performance of two classifiers over all datasets [7]: "Wilcoxon signed-ranks test" and "Counts of wins, losses and ties: Sign test". Using both of these tests, without concentration on details, MIGRW is significantly better than other methods for significance level of 95%.

6.4. **Stability.** In order to measure the noise stability of the proposed method in comparison with others, the class labels of 30% of the training data are changed before the learning process. It is usually expected that the performance of the classifier decreases after inserting the noise. However, in the case of using weak learners such as fuzzy classifiers, because the maximum capability of the classification may not be achieved, experiments on clean datasets may result in worse than in noisy environments on a few datasets. This is why; only the results of R&P, IGARW and MIGRW, which have a good degree of learning power, are compared here. Classification rates of these methods are measured by ten fold-ten cv and are reported in Table 8 for both cases, clean and noisy datasets. Only for "Lung cancer" dataset, the classification results on noisy dataset are to some extent better than on the clean dataset. It should be mentioned that, the initial classification rate of the fuzzy system on this dataset is even lower than the accuracy of a random



(a)



(b)

FIGURE 10. (a) The Artificial Generated Points and the Optimal Decision Boundary (b) Trace of the Margin-based Error vs. ω_2 with Various Margin-sizes

Dataset	No weight	G1	G2	G3	G4	R&P	IGARW	MIGRW	Ttest
Haberman	72.958.8	73.558.0	73.558.0	-	-	73.518.3	72.478.1	76.338.4	+
Pima	65.924.7	65.104.7	65.104.7	-	-	73.204.4	75.063.9	76.773.8	+
Bcancer	94.302.4	88.613.7	88.293.6	-	-	95.952.4	96.462.2	97.172.0	+
WDBC5	92.733.5	77.196.5	78.966.1	-	-	94.413.2	93.663.4	95.352.7	+
WPBC	76.678.5	76.268.2	76.268.2	-	-	76.838.6	76.718.2	81.027.4	+
Ionosphere	73.567.2	73.757.1	73.757.1	-	-	84.876.7	84.785.6	87.275.5	+
Sonar	69.3511.1	61.8612.42	68.3511.6	-	-	75.859.4	75.788.6	78.988.6	+
Parkinson	75.449.5	75.399.6	75.399.6	-	-	82.678.8	81.988.3	85.917.7	+
Promoters	62.9312.4	73.3111.5	73.8911.5	-	-	84.5810.5	87.3410.6	93.846.3	+
Spect(Float)	79.017.4	79.397.3	79.397.3	-	-	72.949.1	73.078.6	81.157.4	+
Iris	94.675.4	92.736.8	95.135.2	95.075.1	94.935.3	95.675.4	95.135.1	97.534.0	+
Wine	92.716.1	84.769.3	89.808.0	88.738.2	92.496.6	95.844.9	95.374.8	97.553.2	+
Lung Cancer	12.0016.9	20.5819.0	16.4217.6	20.6719.0	19.7518.8	35.3324.3	32.9423.9	43.3423.4	+
Heart	49.369.1	51.878.5	51.878.5	51.878.5	51.808.5	48.738.9	49.869.1	53.018.4	+
Glass	56.2211.0	49.4411.5	51.3011.1	49.5311.5	53.0811.9	58.3411.8	60.3112.0	63.5910.6	+
Image	77.109.1	74.299.3	74.389.2	74.249.6	74.489.3	79.148.6	78.978.3	81.078.2	+
Average	71.56	69.88	70.74	70.82	71.22	76.74	76.87	80.62	

TABLE 7. Ten Fold-ten CV Classification Rate After Rule Weighting by G1-4[21], R&P[26], IGARW [34] and MIGRW

classifier, as shown in Table 7. It seems that this dataset is initially full of noise and is ignored in comparisons reported in Table 8.

Dataset	R&P			IGARW			MIGRW		
	Clean	Noisy	Diff.	Clean	Noisy	Diff.	Clean	Noisy	Diff.
Haberman	73.51∓ 8.3	71.40∓ 9.5	2.11	72.47∓ 8.1	68.70∓ 9.6	3.77	76.33∓ 8.4	75.49∓ 7.8	0.84
Pima	73.20∓ 4.4	71.23∓ 5.5	1.97	75.06∓ 3.9	72.67∓ 5.2	2.39	76.77∓ 3.8	75.35∓ 4.6	1.42
Bcancer	95.95∓ 2.4	95.19∓ 2.6	0.76	96.46∓ 2.2	94.99∓ 2.9	1.47	97.17∓ 2.0	96.43∓ 2.2	0.74
WDBC5	94.41∓ 3.2	90.48∓ 5.4	3.93	93.66∓ 3.4	91.70∓ 4.3	1.96	95.35∓ 2.7	93.48∓ 3.8	1.87
WPBC	76.83∓ 8.6	68.98∓ 10.7	7.85	76.71∓ 8.2	66.52∓ 10.8	10.19	81.02∓ 7.4	78.20∓ 7.9	2.82
Ionosphere	84.87∓ 6.7	80.65∓ 7.2	4.22	84.78∓ 5.6	80.69∓ 8.0	4.09	87.27∓ 5.5	84.11∓ 6.7	3.16
Sonar	75.85∓ 9.4	63.30∓ 12.0	12.55	75.78∓ 8.6	64.10∓ 11.9	11.68	78.98∓ 8.6	69.66∓ 10.2	9.32
Parkinson	82.67∓ 8.8	81.14∓ 9.5	1.53	81.98∓ 8.3	80.49∓ 9.2	1.49	85.91∓ 7.7	85.08∓ 8.1	0.83
Promoters	84.58∓ 10.5	68.58∓ 14.9	16.00	87.34∓ 10.6	65.96∓ 14.0	21.38	93.84∓ 6.3	75.68∓ 13.3	18.16
Spect(Float)	72.94∓ 9.1	70.55∓ 9.6	2.39	73.07∓ 8.6	65.98∓ 8.5	7.09	81.15∓ 7.4	79.76∓ 7.7	1.39
Iris	95.67∓ 5.4	92.93∓ 6.7	2.74	95.13∓ 5.1	92.21∓ 6.7	2.92	97.53∓ 4.0	96.67∓ 4.6	0.86
Wine	95.84∓ 4.9	90.36∓ 6.9	5.48	95.37∓ 4.8	89.85∓ 7.0	5.52	97.55∓ 3.2	94.74∓ 5.0	2.81
Lung Cancer	35.33∓ 24.3	37.58∓ 28.4	-	32.94∓ 23.9	33.37∓ 26.3	-	43.34∓ 23.4	43.84∓ 27.8	-
Heart	48.73∓ 8.9	47.99∓ 10.7	0.74	49.86∓ 9.1	48.09∓ 10.6	1.77	53.01∓ 8.4	51.23∓ 10.4	1.78
Glass	58.34∓ 11.8	38.54∓ 15.2	19.8	60.31∓ 12.0	38.48∓ 14.3	21.83	63.59∓ 10.6	42.43∓ 15.5	21.16
Image	79.14∓ 8.6	75.00∓ 10.0	4.14	78.97∓ 8.3	74.11∓ 9.9	4.86	81.07∓ 8.2	78.31∓ 9.5	2.76
Average	76.74	71.49	5.75	76.87	70.49	6.83	80.62	76.28	4.66

Table 8. Comparison of Ten Fold-ten cv Classification Rates on Clean and Noisy Datasets

The difference of classification rates on clean and noisy datasets are also reported in Table 8. The lower difference, the more stability in presence of the noise exists. As shown, MIGRW is more stable than other methods for 12 out of 15 datasets and also in average. This stability is also significantly verified by t-test with confidence level 95%. In addition, the classification rate of the proposed method is still significantly better than other methods in noisy environments.

7. Conclusion

In this paper, a novel margin-based model is proposed for weighting the rules in a fuzzy rule-based classifier. This model is a quadratic convex model inspired by FKSVM, which is modified to support single-winner reasoning to achieve more interpretability. In addition, the cost function is changed to reduce the undesired effects of the noisy(mislabelled) instances. With these modifications, the model is not more convex. This model has been solved by a novel descent optimization method with a low time-complexity and high rule reduction (as a side-effect, which increases interpretability). In addition the significant improvement of the accuracy achieved by this model, it seems that the proposed model is even more stable in presence of the noise. More attention on selecting a proper value of M, improving the cost function, solving the model without discrete approximation and investigating the order of the rules in the weighting process, are some of the works, which can be considered as research topics in the future.

Acknowledgements. M. Taheri thanks Mrs. E. Ashoor Mahani for her help.

References

[1] A. Cano, A. Zafra and S. Ventura, *An EP algorithm for learning highly interpretable classifiers*, 11th Inter. Conf. on Intelligent Systems Design and Applications (ISDA), (2011), 325–330.

[2] S. M. Chen, *Generating weighted fuzzy rules from relational database systems for estimating values using genetic algorithms*, IEEE Trans. on Fuzzy Systems, **11(4)** (2003), 495–506.
[3] S. M. Chen, *A new weighted fuzzy rule interpolation method based on GA-based weights-learning techniques*, proced. of ICMLC, **5** (2010), 2705-2711.
[4] S. M. Chen, *Weighted fuzzy rule interpolation based on GA-based weight-learning techniques*, IEEE Trans. on Fuzzy Systems, **19(4)** (2011), 729–744.
[5] Z. Chi, H. Yan and T. Pham, *Fuzzy algorithms: with applications to image processing and pattern recognition*, World Scientific, Singapore, 1996.
[6] C. Cortes and V. Vapnik, *Support vector networks*, Machine Learning, **20** (2004).
[7] J. Demsar, *Statistical comparisons of classifiers over multiple data sets*, Journal of Machine Learning Research, **7** (2006), 1–30.
[8] S. M. Fakhrahmad and M. Z. Jahromi, *A new rule-weight learning method based on gradient descent*, Proc. of World Congress on Engineering, **I** (2009).
[9] G. Forman and I. Cohen, *Learning from little: comparison of classifiers given little training*, PKDD 2004, LNAI 3202, Springer-Verlag Berlin Heidelberg 2004, 2004, 161–172.
[10] L. Fu, *Rule generation from neural networks*, IEEE Transaction on systems, Man, and Cybernetics, **24(8)** (1994).
[11] S. Garcia and F. Herrera, *An extension on statistical comparison of classifiers over multiple data sets for all pair wise comparisons*, Journal of Machine Learning Research, **9** (2008), 2677–2694.
[12] C. Hsu and C. Lin, *A comparison of methods for multiclass support vector machines*, IEEE Transaction on neural networks, **13(2)** (2002).
[13] Q. Hu, P. Zhu, Y. Yang and D. Yu, *Large-margin nearest neighbor classifiers via sample weight learning*, Neurocomputing, **74(4)** (2011), 656–660.
[14] H. Ishibuchi, T. Murata and I. B.Turksen, *Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems*, Fuzzy Sets and Systems, **89(2)** (1997), 135–150.
[15] H. Ishibuchi and T. Nakashima, *Effect of rule weights in fuzzy rule-based classification systems*, IEEE Transactions on Fuzzy Systems, **9(4)** (2001), 506–515.
[16] H. Ishibuchi, T. Nakashima and M. Nii, *Classification and modelling with linguistic information granules: advanced approaches to linguistic data mining*, Springer Verlag, 2004.
[17] H. Ishibuchi and M. Nii, *Techniques and applications of neural networks for fuzzy rule approximation*, Fuzzy Theory Systems, (1999), 1491–1519.
[18] H. Ishibuchi and Y. Nojima, *Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning*, International Journal of Approximate Reasoning, **44(1)** (2007), 4–31.
[19] H. Ishibuchi, K. Nozaki and H. Tanaka, *Distributed representation of fuzzy rules and its application to pattern classification*, Fuzzy Sets and Systems, **52(1)** (1992),21–32.
[20] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, *Selecting fuzzy if-then rules for classification problems using genetic algorithms*, IEEE Transactions on Fuzzy Systems, **3(3)** (1995),260–270.
[21] H. Ishibuchi and T. Yamamoto, *Rule weight specification in fuzzy rule-based classification systems*, IEEE Trans. on Fuzzy Systems, **13(4)** (2005), 428–435.
[22] 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, **141(1)** (2004), 59–88.
[23] J. Langford, *Tutorial on practical prediction theory for classification*, Journal of Machine Learning Research, **6** (2005), 273–306.
[24] R. Mikut, J. Jakel and L. Groll, *Interpretability issues in data-based learning of fuzzy systems*, Elsevier, Fuzzy Sets and Systems, **150** (2005), 179–197.
[25] T. Nakashima, G. Schaefer, Y. Yokota and H. Ishibuchi, *A weighted fuzzy classifier and its application to image processing tasks*, Fuzzy Sets and Systems, **158(3)** (2007), 284–294.
[26] K. Nozaki, H. Ishibuchi and H. Tanaka, *Adaptive fuzzy rule-based classification systems*, IEEE Transactions on Fuzzy Systems, **4(3)** (1996), 238–250.

[27] M. Taheri, H. Azad, K. Ziarati and R. Sanaye, *A quadratic margin-based model for weighting fuzzy classification rules inspired by support vector machines*, Iranian Journal of Fuzzy Systems, **10(4)** (2013), 41–55.
[28] *UCI machine learning repository, http://www.ics.uci.edu/ mlearn/databases.*
[29] V. Vapnik, *The nature of statistical learning theory*, Springer Verlag, New York, 1995.
[30] K. Q. Weinberger, J. C. Blitzer and L. K. Saul, *Distance metric learning for large margin nearest neighbor classification*, Advances in Neural Information Processing Systems (NIPS), **18** (2006), 1473–1480.
[31] J. Weston and C. Watkins, *Support vector machines for multi-class pattern recognition*, ESANN'1999 proceedings - European Symposium on Artificial Neural Networks, Bruges (Belgium), isbn:2-600049-9-X, (1999), 219–224.
[32] L. Xu, K. Crammer and D. Schuurmans, *Robust support vector machine training via convex outlier ablation*, In Proc. of the National Conf. on Artificial Intelligence, **21(1)** (2006), 536–542.
[33] L. Yu and J. Xiao, *Trade-off between accuracy and interpretability: experience-oriented fuzzy modeling via reduced-set vectors*, Elsevier, Computers and Mathematics with Applications, **57** (2009), 885–895.
[34] M. J. Zolghadri and M. Taheri, *A proposed method for learning rule weights in fuzzy rule-based classification systems*, Fuzzy Sets and Systems, **159** (2008), 449–459.

Mohammad Taheri*, Computer Science & Engineering & IT Department of Shiraz University, Shiraz, Fars, Iran
E-mail address: mtaheri@cse.shirazu.ac.ir

Mansoor Zolghadri Jahromi, Computer Science & Engineering & IT Department of Shiraz University, Shiraz, Fars, Iran
E-mail address: zjahromi@shirazu.ac.ir

*Corresponding author