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
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
A Margin-based Model with a Fast Local Search for Rule Weighting and Reduction in ...

An 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:

\[ \text{Rule}_{r_k} : \text{If } x^1 = A^1_k \land x^2 = A^2_k \land \ldots \land x^d = A^d_k \Rightarrow \text{class } h_k \text{ with } (\omega_k) \] (1)

Where, \( r_k \) denotes \( k^{th} \) rule in the rule-base, \( x^i \) is the \( i^{th} \) feature value of the input pattern \( X = [x^1, x^2, \ldots, x^d] \), \( d \) is the number of features, \( A^i_k \) 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 \( \omega_k \) 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:

\[ \mu_k(X) = \prod_{1 \leq i \leq d} \eta(x^i, A^i_k) \] (2)

Where \( \mu_k(X) \) denotes the compatibility grade of instance \( X \) with \( r_k \) and \( \eta(x^i, A^i_k) \) denotes the degree of the membership of \( x^i \) in the fuzzy set \( A^i_k \). 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:

\[
X \text{ is classified as } h_k^* \text{ where } k^* = \arg\max_k [\omega_k \mu_k(X)] \tag{3}
\]

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:

\[
X \text{ is classified as } c^* \in CL \text{ where } c^* = \arg\max_c \left[ \sum_{r_k \in R^c} \omega_k \mu_k(X) \right] \tag{4}
\]

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, \( \omega_1 \) 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, \( \omega_2 \) is set to its optimal value considering other rule weights (even \( \omega_1 \) with its new value) are fixed. This process is done for all the weights in \( |R| \) steps. It is possible that, optimality of \( \omega_1 \) 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 \( \omega_k \) are given fixed and it is desired to set \( \omega_k \) 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 ($\epsilon_j$) 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).

$$\begin{align*}
\text{minimize } \quad E &= \sum_{X_j} \ell(\epsilon_j) \\
\text{s.t. } \forall X_j : \quad \epsilon_j &= \max_{r_i \in R^{y_j}} (\omega_i \mu_i(X_j)) - \max_{r_i \in R^{y_j}} (\omega_i \mu_i(X_j)) \\
\ell(\epsilon_j) &= \begin{cases} 
0 & \text{if } \epsilon_j < 0 \\
1 & \text{if } \epsilon_j \geq 0 
\end{cases}
\end{align*}$$

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 $\ell(\epsilon_j)$ 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 $\omega_k$, 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. $\mu_k(X) = 0$), the value of $\omega_k$ does not effect on the classification result of $X$ and it is ignored here. Otherwise, by increasing the value of $\omega_k$, 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).

$$th^{cut} = \max_{r_i \in R^{y_j}} \left[ \frac{\omega_i \mu_i(X_j)}{\mu_k(X_j)} \right]$$

For $\omega_k < th^{cut}$, the distance between $X_j$ and the decision boundary does not change. However, for values of $\omega_k$ 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).

$$th = \max_{r_i \in R^{y_j}} \left[ \frac{\omega_i \mu_i(X_j)}{\mu_k(X_j)} \right]$$

If $th$ is less than $th^{cut}$, increasing $\omega_k$ 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.
Setting $\omega_k$ to any value from the optimal interval optimizes the objective function respect to $\omega_k$. In IGARW, if the optimal interval is between two consecutive thresholds $(th^j, th^{j+1})$, $\omega_k$ is set to the mean of its values i.e. $(th^j + th^{j+1})/2$. If the intervals $[0, th^1)$ or $(th^n, +\infty)$ are optimal, $\omega_k$ will be set to 0 or $th^n + \tau$, respectively (where, $\tau$ 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).

$$\begin{align*}
\text{Maximize} & \quad M \\
\text{s.t.} & \quad \forall X_j, c \neq y_j : f_{c,y_j}(X_j) \geq M
\end{align*}$$

(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.

$$\begin{align*}
\text{Maximize} & \quad M \\
\text{s.t.} & \quad \forall X_j : \frac{y_j (W^T X_j + b)}{\|W\|} \geq M \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad y_j \in \{+1, -1\}
\end{align*}$$

(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).

$$\begin{align*}
\text{Maximize} & \quad \frac{1}{\|W\|} \\
\text{or} & \\
\text{Minimize} & \quad \frac{1}{2} W^2 \\
\text{s.t.} & \quad \forall X_j : \frac{y_j (W^T X_j + b)}{\|W\|} \geq 1 \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad y_j \in \{+1, -1\}
\end{align*}$$

(10)
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).

$$\text{Maximize} \quad M$$

$$s.t. \quad \forall X_j, c \neq y_j : \quad \sum_{r_i \in R^y_j} \omega_i \mu_i(X_j) - \sum_{r_i \in R^c} \omega_i \mu_i(X_j) \geq M$$

$$\omega_i \geq 0$$ (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.

$$\text{Maximize} \quad M$$

$$s.t. \quad \forall X_j : \quad \frac{\max_{r_i \in R^y_j} \omega_i \mu_i(X_j) - \max_{r_i \in R^c} \omega_i \mu_i(X_j)}{\|\Psi\|} \geq M$$

$$\omega_i \geq 0$$ (12)
Due to have non-negativity constraints on the rule-weights ($\omega_i \geq 0$), the first norm of $\Psi$ 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).

\[
\begin{align*}
\text{Minimize} & \quad \sum_{X_j} e_j \\
\text{s.t.} & \quad \forall X_j: \max_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) - \max_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) - \sum \omega_i \geq M - e_j \\
& \quad \omega_i \geq 0 \quad e_j \geq 0
\end{align*}
\]

(13)

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

\[
\begin{align*}
\text{Minimize} & \quad \sum_{X_j} \Gamma(e_j) \\
\text{s.t.} & \quad \forall X_j: \quad e_j = \max_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) - \max_{r_i \in R^{y_j}} \omega_i \mu_i(X_j) \\
& \quad \omega_i \geq 0
\end{align*}
\]

\begin{equation}
\Gamma(\epsilon) = \begin{cases} 
0 & \text{if } \epsilon < -M \\
\epsilon + M & \text{if } \epsilon \geq -M
\end{cases}
\end{equation}

(14)

This model is similar to (5). Each training instance has a relative error $e_j$ that produces some cost in the objective function using the non-decreasing cost function $\Gamma$. 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](image-url)
A Margin-based Model with a Fast Local Search for Rule Weighting and Reduction in ...

\[ \Gamma^{FL}(\epsilon) = \begin{cases} 
0 & \text{if } \epsilon < -M \\
(\epsilon + M)/2M & \text{if } -M \leq \epsilon \leq M \\
1 & \text{if } M < \epsilon
\end{cases} \] (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.

\[ \tilde{\Gamma}^p(\epsilon) = \frac{1}{2p + 1} \sum_{t=-p}^{p} \ell \left( \epsilon - t \frac{M}{p} \right) \] (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 \( \omega_k \) optimizes the following model respect to only \( \omega_k \).

\[
\begin{align*}
\text{Minimize} & \quad \sum_{X_j} \sum_{t=1}^{p} \omega_t \left( \sum_{h \in H} \omega_t \mu_t(X_j) - \sum_{i \in H} \omega_i \mu_i(X_j) \right) \\
\text{s.t.} & \quad \forall X_j, \quad \omega_t \geq 0
\end{align*}
\] (17)

By increasing \( \omega_k \), the weighted compatibility grade of \( X_j \) with \( r_k \) increases. There are some threshold values \( th \) whereas for \( \omega_k \) 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 \( \omega_k \) is divided into some intervals, in which the objective function is fixed. It is desired to find the optimal interval for values of \( \omega_k \). If \( \omega_k \) 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 \( \omega_i \). A cost +1 or -1 is attached to a threshold if passing \( \omega_k \) from that threshold, the objective function increases or decreases, respectively. Each step function is identified by associated training instance \( X \) and the value of \( t' = \frac{tM}{p} \) 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 \( \omega_k \) 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).

\[
\begin{align*}
\text{th}_1 = \frac{\max_{r \in H^X} \omega_i \mu_i(X) - \max_{r \in H^X} \omega_i \mu_i(X)}{t'} - \sum_{i \in H^X} \omega_i
\end{align*}
\] (18)

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

\[
\begin{align*}
\text{th}_2 = \frac{\max_{r \in H^X} (\omega_i \mu_i(X)) - \sum_{i \in H^X} \omega_i}{t' + \mu_k(X)}
\end{align*}
\] (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).

\[
\begin{align*}
\text{th}_2 = \frac{\max_{r \in H^X} (\omega_i \mu_i(X)) + \sum_{i \in H^X} \omega_i}{-t' + \mu_k(X)}
\end{align*}
\] (20)

Acceptable thresholds \( th_1 \) and \( th_2 \) for each step function are stored in a list and the optimal value of \( \omega_k \) 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.
1: function Rule_Weighting(R, T, M, p)
2:     ▷ The set of fuzzy rules \( R = \{r_1, r_2, \ldots, r_{|R|}\} \)
3:     ▷ The training set \( T = \{X_1, X_2, \ldots, X_{|T|}\} \)
4:     ▷ \( M \) as the margin-size and \( p \) as the approximation parameter
5:     ∀\( r_k : \omega_k \leftarrow 1 \) ▷ Initialize all the rule weights to 1
6:     for I iterations do
7:         for each \( r_k \in R \) do
8:             L ← an empty set
9:             for each \( X \in T \) do
10:                for \( t \leftarrow -p \) to \( p \) do
11:                   \( t' \leftarrow t \frac{M}{p} \)
12:                      compute \( th^{cut}_1, th_1 \) and associated \( cost_1 \)
13:                      if \( 0 \leq th_1 < th^{cut} \) then
14:                          insert \( (th_1, cost_1) \) to \( L \)
15:                      end if
16:                      compute \( th^{cut}_2, th_2 \) and associated \( cost_2 \)
17:                      if \( th_2 > th^{cut} \) then
18:                          insert \( (th_2, cost_2) \) to \( L \)
19:                      end if
20:                  end for
21:         end for
22:     end for
23:     Sort \( L \) in ascending order respect to threshold values.
24:     ▷ \( L = [\langle th^i, cost^i \rangle \mid 1 \leq i \leq n \} \) where \( i < j \rightarrow th^i \leq th^j \)
25:     Find the best interval in \( L \)
26:     if best interval=\([0, th^1)\) then
27:         \( \omega_k \leftarrow 0 \)
28:     else if best interval=(\(th^n, +\infty\) then
29:         \( \omega_k \leftarrow \omega_k + \tau \)
30:     else if best interval=\((th^i, th^{i+1})\) then
31:         \( \omega_k \leftarrow \frac{th^i + th^{i+1}}{2} \)
32:     end if
33:     end for
34: return \( \Psi \)
35: 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-approximation. 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.
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.

Table 2. List of UCI Datasets Used in This Paper

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Haberman</td>
<td>306</td>
<td>3</td>
<td>2</td>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Pima</td>
<td>768</td>
<td>8</td>
<td>2</td>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Bcancer</td>
<td>684</td>
<td>10</td>
<td>2</td>
<td>Lung</td>
<td>32</td>
<td>56</td>
<td>3</td>
</tr>
<tr>
<td>WDBC5</td>
<td>569</td>
<td>30</td>
<td>2</td>
<td>Cancer</td>
<td>287</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>WPBC</td>
<td>198</td>
<td>33</td>
<td>2</td>
<td>Heart</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
<td>Glass</td>
<td>210</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Sonar</td>
<td>208</td>
<td>60</td>
<td>2</td>
<td>Image</td>
<td>210</td>
<td>19</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 5. Fourteen Fuzzy Sets Used in This Paper for Rule Generation
| Dataset       | No Weight | G1 Rate | G1 Rule# | G2 Rate | G2 Rule# | G3 Rate | G3 Rule# | G4 Rate | G4 Rule# | R&P Rate | R&P Rule# | IGARW Rate | IGARW Rule# | MIGRW Rate | MIGRW Rule# |
|--------------|-----------|---------|----------|---------|----------|---------|----------|---------|----------|----------|-----------|-----------|------------|------------|------------|------------|
| Haberman     | 73.53     | 60      | 74.18    | -       | -        | 74.18   | -        | 77.45   | -        | 80.72    | 9         | 73.86      | 60         | 75.00      | 55         |
| Pima         | 65.89     | 60      | 69.06    | 69.66   | -        | -       | 72.49    | 79.04   | -        | 75.00    | 55        | 94.02      | 13         | 94.02      | 13         |
| BCancer      | 94.01     | 60      | 94.30    | 94.30   | -        | -       | 96.35    | 98.10   | 6        | 96.93    | 4         | 94.02      | 13         | 94.02      | 13         |
| WDBC5        | 93.15     | 60      | 93.67    | 93.32   | -        | -       | 93.85    | 96.84   | 6        | 94.02    | 13        | 94.02      | 13         | 94.02      | 13         |
| WPBC         | 77.27     | 60      | 77.27    | 77.27   | -        | -       | 82.32    | 86.36   | 9        | 84.34    | 8         | 87.75      | 25         | 87.75      | 25         |
| Ionosphere   | 67.52     | 60      | 86.61    | 87.18   | -        | -       | 88.03    | 89.46   | 6        | 87.75    | 25        | 87.75      | 25         | 87.75      | 25         |
| Sonar        | 70.47     | 60      | 75.44    | 77.40   | -        | -       | 68.87    | 83.17   | 8        | 77.88    | 22        | 87.69      | 8          | 87.69      | 8          |
| Parkinson    | 75.38     | 60      | 75.38    | 75.38   | -        | -       | 83.59    | 89.23   | 4        | 87.69    | 8          | 87.69      | 8          | 87.69      | 8          |
| Promoters    | 68.87     | 60      | 72.64    | 73.58   | -        | -       | 92.45    | 97.17   | 8        | 96.23    | 8          | 96.23      | 8          | 96.23      | 8          |
| Spect(Float) | 79.40     | 60      | 79.40    | 79.40   | -        | -       | 83.52    | 91.01   | 14       | 80.15    | 7          | 80.15      | 7          | 80.15      | 7          |
| Iris         | 94.67     | 90      | 95.33    | 96      | 95.33    | 96      | 99.33    | 98.67   | 5        | 96.67    | 3          | 96.67      | 3          | 96.67      | 3          |
| Wine         | 94.94     | 90      | 96.07    | 95.31   | 95.51    | 96.07   | 100      | 100     | 8        | 97.75    | 9          | 97.75      | 9          | 97.75      | 9          |
| Lung Cancer  | 53.13     | 90      | 96.88    | 93.75   | 96.88    | 96.88   | 84.38    | 100     | 9        | 96.88    | 7          | 96.88      | 7          | 96.88      | 7          |
| Heart        | 57.14     | 150     | 60.28    | 61.67   | 60.28    | 61.67   | 58.89    | 66.90   | 16       | 55.75    | 139        | 55.75      | 139        | 55.75      | 139        |
| Glass        | 62.61     | 130     | 62.61    | 65.42   | 62.61    | 65.42   | 62.61    | 65.42   | 13       | 64.02    | 9          | 64.02      | 9          | 64.02      | 9          |
| Image        | 77.62     | 210     | 79.52    | 79.05   | 80.00    | 78.10   | 81.43    | 81.43   | 8        | 79.52    | 11         | 79.52      | 11         | 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      | 84.03      | 24.25      | 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

<table>
<thead>
<tr>
<th>Rule #</th>
<th>If x_3 is very small then class 1 with confidence 92.64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule #</td>
<td>If x_3 is not small or large &amp; x_0 is medium then class 2 with confidence 176.7</td>
</tr>
<tr>
<td>Rule #</td>
<td>If x_2 is large then class 3 with confidence 167.8</td>
</tr>
</tbody>
</table>

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](image)

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-dimensional 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 \(\omega_2\).

| Rule # 1 | If \(x\) is small then class + with confidence 1 |
| Rule # 2 | If \(x\) is medium then class * with confidence \(\omega_2\) |
| 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 \(\omega_2\) less than or greater than 0.43, one of these instances are misclassified. In weighting \(\omega_2\), 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 \(\omega_2\) 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. \(\omega_2\) in Figure 8.

![Figure 7. Decision Spaces of the Rules in Table 5 on a 1-dimensional Dataset with Different Weights of the 2nd Rule:](image)

\(\omega_2 = 2.33\) (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
margin-size, the optimal interval converges to the point $\omega_2=0.82$ 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 $\omega_2=0.82$. 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.

In the former experiment, an artificial 2-dimensional 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 $A$ with confidence $1$.

Rule #2: If $x^1$ is not small & $x^2$ is not small then class $O$ with confidence $\omega_2$.

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 $\omega_2=1$. However, as depicted in Figure 10(b), with IGARW (i.e. M=0) the optimal value of $\omega_2$ is 2.36. By increasing the margin-size up to M=0.4, the optimum value of $\omega_2$ 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 $1/16 \approx 0.063$. 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
Figure 10. (a) The Artificial Generated Points and the Optimal Decision Boundary (b) Trace of the Margin-based Error vs. $\omega^2$ with Various Margin-sizes

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No weight</th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>G4</th>
<th>R&amp;P</th>
<th>IGARW</th>
<th>MIGRW</th>
<th>Ttest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haberman</td>
<td>72.9</td>
<td>73.5</td>
<td>73.5</td>
<td>-</td>
<td>-</td>
<td>73.5</td>
<td>75.0</td>
<td>76.7</td>
<td>+</td>
</tr>
<tr>
<td>Pima</td>
<td>65.9</td>
<td>65.1</td>
<td>64.7</td>
<td>73</td>
<td>73.5</td>
<td>73.5</td>
<td>74.4</td>
<td>74.4</td>
<td>+</td>
</tr>
<tr>
<td>Breast</td>
<td>94.3</td>
<td>94.3</td>
<td>94.3</td>
<td>-</td>
<td>-</td>
<td>94.3</td>
<td>94.3</td>
<td>94.3</td>
<td>+</td>
</tr>
<tr>
<td>WDBC4</td>
<td>92.7</td>
<td>92.7</td>
<td>92.7</td>
<td>-</td>
<td>-</td>
<td>92.7</td>
<td>92.7</td>
<td>92.7</td>
<td>+</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>73.6</td>
<td>73.6</td>
<td>73.6</td>
<td>-</td>
<td>-</td>
<td>73.6</td>
<td>73.6</td>
<td>73.6</td>
<td>+</td>
</tr>
<tr>
<td>Sonar</td>
<td>69.3</td>
<td>69.3</td>
<td>69.3</td>
<td>-</td>
<td>-</td>
<td>69.3</td>
<td>69.3</td>
<td>69.3</td>
<td>+</td>
</tr>
<tr>
<td>Parkinson</td>
<td>75.4</td>
<td>75.4</td>
<td>75.4</td>
<td>-</td>
<td>-</td>
<td>75.4</td>
<td>75.4</td>
<td>75.4</td>
<td>+</td>
</tr>
<tr>
<td>Promoters</td>
<td>82.3</td>
<td>82.3</td>
<td>82.3</td>
<td>-</td>
<td>-</td>
<td>82.3</td>
<td>82.3</td>
<td>82.3</td>
<td>+</td>
</tr>
<tr>
<td>Spect(Float)</td>
<td>79.4</td>
<td>79.4</td>
<td>79.4</td>
<td>-</td>
<td>-</td>
<td>79.4</td>
<td>79.4</td>
<td>79.4</td>
<td>+</td>
</tr>
<tr>
<td>Iris</td>
<td>94.3</td>
<td>94.3</td>
<td>94.3</td>
<td>-</td>
<td>-</td>
<td>94.3</td>
<td>94.3</td>
<td>94.3</td>
<td>+</td>
</tr>
<tr>
<td>Wine</td>
<td>92.7</td>
<td>92.7</td>
<td>92.7</td>
<td>-</td>
<td>-</td>
<td>92.7</td>
<td>92.7</td>
<td>92.7</td>
<td>+</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>12.0</td>
<td>12.0</td>
<td>12.0</td>
<td>-</td>
<td>-</td>
<td>12.0</td>
<td>12.0</td>
<td>12.0</td>
<td>+</td>
</tr>
<tr>
<td>Heart</td>
<td>49.3</td>
<td>49.3</td>
<td>49.3</td>
<td>-</td>
<td>-</td>
<td>49.3</td>
<td>49.3</td>
<td>49.3</td>
<td>+</td>
</tr>
<tr>
<td>Glass</td>
<td>56.2</td>
<td>56.2</td>
<td>56.2</td>
<td>-</td>
<td>-</td>
<td>56.2</td>
<td>56.2</td>
<td>56.2</td>
<td>+</td>
</tr>
<tr>
<td>Image</td>
<td>77.1</td>
<td>77.1</td>
<td>77.1</td>
<td>-</td>
<td>-</td>
<td>77.1</td>
<td>77.1</td>
<td>77.1</td>
<td>+</td>
</tr>
<tr>
<td>Average</td>
<td>71.5</td>
<td>70.8</td>
<td>70.8</td>
<td>70.8</td>
<td>70.8</td>
<td>70.8</td>
<td>70.8</td>
<td>70.8</td>
<td>70.8</td>
</tr>
</tbody>
</table>

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

References


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