Robust Support Vector Machine-Trained Fuzzy System

Yahya Forghani  
yahyafor2000@yahoo.com,  
ya_yo365@stu.um.ac.ir  
1- Department of Computer Engineering,  
Ferdowsi University of Mashhad  
2- Department of Computer Engineering,  
Islamic Azad University, Mashhad Branch  
Mashhad, Iran

Hadi Sadoghi Yazdi,  
h-sadoghi@um.ac.ir  
1- Department of Computer Engineering,  
Ferdowsi University of Mashhad  
2- Center of Excellence on Soft Computing  
and Intelligent Information Processing,  
Ferdowsi University of Mashhad  
Mashhad, Iran

Abstract: Because the SVM (support vector machines) classifies data with the widest symmetric margin to decrease the probability of test error, modern fuzzy systems use SVM to tune the parameters of fuzzy if-then rules. But, solving the SVM model is time-consuming. To overcome this disadvantage, we propose a rapid method to solve the robust SVM model and use it to tune the parameters of fuzzy if-then rules. The robust SVM is an extension of SVM for interval-valued data classification.

We compare our proposed method with SVM, robust SVM, ISVM-FC (incremental support vector machine-trained fuzzy classifier), BSVM-FC (batch support vector machine-trained fuzzy classifier), SOTFN-SV (a self-organizing TS-type fuzzy network with support vector learning) and SCLSE (a TS-type fuzzy system with subtractive clustering for antecedent parameter tuning and LSE for consequent parameter tuning) by using some real datasets. According to experimental results, the use of proposed approach leads to very low training and testing time with good misclassification rate.

Index Terms— Fuzzy system; Classification; robust SVM; Rule.
I. INTRODUCTION

A fuzzy system can be used for data classification [1-5]. The parameters of fuzzy rules in such fuzzy system (or the parameter of classifier) may be learned by minimizing only the training error. There are many different classifiers with the same training error for a classification problem. But one of them classifies data with the widest symmetric margin. This classifier has the lowest test error if both training and test samples are from identical probability distribution [6-8]. Such classifier is called optimal classifier [9]. Minimizing only the training error may not lead us to the optimal classifier. The SVMs can find the mentioned optimal classifier namely a classifier with the widest symmetric margin. Y. Chen and J. Z. Wang showed that SVMs is a fuzzy system [1] and named it positive definite fuzzy classifier (PDFC). Each support vectors (SVs) in Gaussian-kernel SVMs can be interpreted as the center of the antecedent part fuzzy set of a fuzzy if-then rule in PDFC. The number of Support Vectors (SVs) in SVMs and thereby the number of rules in PDFC is usually high which leads us to a classifier with high test time. Meanwhile, the consequent part parameters of fuzzy if-then rules are learned by solving SVMs model which is time-consuming.

Similar to PDFC, the support-vector-based fuzzy neural network (SVFNN) [2] builds initial rules from SVs. It then uses a learning algorithm to remove less important fuzzy rules. This rule reduction approach does not maintain the optimal classifier and thereby may increase the probability of test error. Fuzzy system learned through fuzzy clustering and SVM (FS-FCSVM) [5] uses a clustering strategy to decrease the number of rules. The number of rules in FS-FCSVM depends on the number of fuzzy clusters instead of the number of SVs. The number of clusters and thereby the number of rules is tunable. Therefore, the number of rules can be decreased if a low test time is desired.
PDFC, SVFNN and FS-FCSVM use zero-order Takagi–Sugeno (TS)-type fuzzy rules. The performance of a fuzzy system with first-order TS-type consequent part is better than that with zero-order TS-type consequent part [10]. Therefore, three different subsequent fuzzy system, i.e. Self-organizing TS-type fuzzy network with support vector learning (SOTFN-SV) [4], batch support vector machine-trained fuzzy classifier (BSVM-FC) and incremental support vector machine-trained fuzzy classifier (ISVM-FC) [3] used first-order TS-type fuzzy if-then rule.

Contrary to SOTFN-SV, all of the free consequent part parameters in each TS-type fuzzy rule of ISVM-FC and BSVM-FC are tunable. ISVM-FC is suitable for time-dependent classification problems where a subset of training data is available at any given time because it uses an incremental SVM instead of a batch SVM to tune the consequent part parameters of fuzzy if-then rules.

In each of the mentioned SVM-based fuzzy systems, an SVM model is solved. Solving the SVM model is time consuming. To overcome this disadvantage, we propose a rapid method to solve the robust SVM [11] model and use it to tune the consequent part parameters. The robust SVM is a model for interval valued data classification with the widest symmetric margin. To use this model for real valued data classification, first, we merge each set of real valued neighbors with the same class label to obtain a small interval-valued training set. Then, we construct a TS-type rule for each interval-valued data based on the robust SVM model. Therefore, the number of rules in the proposed method is equal to the number of interval-valued data which is often much lower than the number of Gaussian-kernel support vectors in SVM or the number of rules in PDFC.

We compare our proposed method with (a) PDFC (or SVM), (b) robust SVM (c) SCLSE[12], (d) SOTFN-SV, (e) BSVM-FC and (f) ISVM-FC, by using some real datasets. According to
experimental results, the use of proposed method leads to very low training and test time with good misclassification rate.

The organization of this paper is as follows: In section II, we pay to preliminaries. The paper then proceeds to explain our novel approach in section III. Section IV compares time complexity of our proposed method with the other classification methods. In section V, we show some numerical examples. In section VI, we apply our novel method for the classification of some real datasets. Finally, in section VII, we set out our conclusions.

II. PRELIMINARIES

A. TS-type Fuzzy Inference System

Consider the following \( l \) TS-type fuzzy if-then rules:

Rule \( i \): If \( X_1 \) is \( A_{i_1} \) and \( X_2 \) is \( A_{i_2} \) and … and \( X_p \) is \( A_{i_p} \), then

\[
f(X) = w_i^T X + b_i, \quad i = 1,2,\ldots,l;
\]

where \( w_i = (w_{i1}, w_{i2}, \ldots, w_{ip}) \) is a weight vector, \( p \) is the dimension of \( w_i \), \( b_i \) is a bias, \( X \in \mathbb{R}^p \) and \( A_{ij} \) is a fuzzy set. The \( w_i \) and \( b_i (i = 1,2,\ldots,l) \) are called consequent parameters and if for example each \( A_{ij} (i = 1,2,\ldots,l; j = 1,2,\ldots,p) \) is a fuzzy set with a Gaussian membership function, namely \( \mu_{A_{ij}}(X) = \exp \left( -\frac{\|X - c_{ij}\|^2}{2\sigma_{ij}^2} \right) \), then \( c_{ij} \) and \( \sigma_{ij} \) (which are the center and standard deviation of the Gaussian membership function, respectively) are called antecedent parameters.

The \( f(\cdot) \), given a crisp input \( X \) (or defuzzified fuzzy input), can be obtained by using Take-winner-all response [13] as follows:
\[ f(X) = f_h(X), \]

where \( f_h(X) = w_h^T X + b_h \), and \( h = \arg \max \left\{ \prod_{j=1}^{p} \mu_{h_j}(X_j) \right\} \).

### B. Support Vector Machines

Formulation of the hard-margin SVM for two-class classification is as follows:

\[
\begin{align*}
\min_{w,b} & \quad \frac{1}{2} \|w\|^2 \\
\text{subject to} & \quad y_i (w^T \varphi(x_i) + b) \geq 1, \quad i = 1, 2, \ldots, n;
\end{align*}
\]

where \( n \) is the number of training samples, \( x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T \) is a \( p \)-dimensional crisp training sample in the input space, \( y_i \in \{1, -1\} \) is the class label of \( x_i \) which has been determined in advance in the dataset and \( \varphi(X) \) is a mapping function that maps \( X \) into a high dimensional feature space in which the classes are linearly separable. In other words, the SVM finds a hyperplane in the high dimensional feature space, namely \( w^T \varphi(X) + b = 0 \), such that the samples are classified correctly, namely

\[
w^T \varphi(x_i) + b \begin{cases} 
\geq 1 \quad \text{for } y_i = +1; \\
\leq 1 \quad \text{for } y_i = -1;
\end{cases}
\]

or equivalently

\[
y_i (w^T \varphi(x_i) + b) \geq 1, \quad i = 1, 2, \ldots, n;
\]

where \( w = (w_1, w_2, \ldots, w_h)^T \) is a weight vector, \( b \) is a bias term and \( h \) is the dimension of the high dimensional feature space. There are an infinite number of hyperplane that satisfy Eq. (3). Fig. 1 shows two hyperplanes that satisfy Eq. (3) for \( \varphi(X) = X \). The generalization ability
depends on the location of the separating hyperplane. The hyperplane with the maximum or the widest symmetric margin is called optimal hyperplane. The symmetric margin of a hyperplane is equal to $\frac{1}{\|w\|^2}$. The model (2) minimizes $\|w\|^2$ or equivalently maximizes the symmetric margin.

Fig. 1. A separating hyperplane that satisfies Eq. (3) and optimal separating hyperplane obtained by the hard-margin SVM for $\varphi(x) = x$.

Formulation of the soft-margin SVM for two-class classification is as follows:

$$
\min_{w, b, \xi_i} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i,
$$

subject to

$$
\begin{align*}
    y_i \left( w^T \varphi(x_i) + b \right) &\geq 1 - \xi_i, &i = 1, 2, ..., n; \\
    \xi_i &\geq 0, &i = 1, 2, ..., n;
\end{align*}
$$

(4)

where $\xi = (\xi_1, \xi_2, ..., \xi_n)^T$, $\xi_i$ is the slack variable of $i$-th training sample, $C$ is a penalty term that determines the trade-off between the maximization of the margin between two classes, namely $\frac{1}{\|w\|^2}$, and minimization of the classification error, namely $\sum_{i=1}^{n} \xi_i$ (For more information see [6-9, 14]). Fig. 2 shows the optimal hyperplane and the slack variables for a classification problem.
The Lagrangian dual form of the model (4) can be restated as follows:

$$\begin{align*}
\max_{\beta_i} \sum_{i=1}^{n} \beta_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j y_i y_j K(x_i, x_j) \\
\text{subject to} \quad \sum_{i=1}^{n} \beta_i y_i = 0; \\
0 \leq \beta_i \leq C, \quad i = 1, 2, ..., n;
\end{align*}$$

(5)

where $\beta_i (i = 1, 2, ..., n)$ are Lagrangian multipliers, and $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ is a kernel function such as Gaussian kernel function, i.e. $K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$, where $\sigma$ is a constant. By using KKT conditions, if $0 \leq \beta_i \leq C$, $b = y_i - w^T \varphi(x_i)$. Then, the decision function for the test sample $X$ is given by $\text{sign}(f(X))$, where from KKT conditions

$$f(X) = w^T \varphi(X) + b = \sum_{i=1}^{n} \beta_i y_i K(x_i, X) + b.$$  

(6)

In [1], it has been shown that Eq. (6) is an especial fuzzy inference system with $l \leq n$ rules, where $l$ is the number non-zero $\beta_i$ (Support Vectors).
As can be seen, the SVM has been designed to learn the parameters of optimal hyperplane according to crisp training data. In continue, we review an extension of the SVM, i.e. robust SVM, proposed for interval-valued data classification. We use this model for tuning the consequent parameters of fuzzy if-then rules in our novel fuzzy system.

C. Robust SVM

In [11], the SVM was extended for robust classification of interval-valued training samples in the input space. We use their proposed strategy for tuning the consequent parameters of fuzzy if-then rules in our novel fuzzy system. Formulation of the robust SVM is as follows:

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & \quad \min_{\tilde{y}} y_i (w^T \tilde{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., n; \\
& \quad \xi_i \geq 0, \quad i = 1, 2, ..., n;
\end{align*}
\]

(7)

where \( \tilde{x}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, ..., \tilde{x}_{ip})^T \) is a training sample whose features are interval values. Each interval-valued data is shown by a hypercube. If all points of \( \tilde{x}_i \) can be classified correctly, \( \arg\min_{\tilde{y}} y_i (w^T \tilde{x}_i + b) \) finds the nearest point of the hypercube of \( \tilde{x}_i \) to the separating hyperplane \( w^T X + b = 0 \), else it finds the farthest point of the hypercube of \( \tilde{x}_i \) which is not classified correctly. We name this point as “worst point of \( \tilde{x}_i \)”. Therefore, the model (7) finds a hyperplane with the widest symmetric margin with respect to the worst points of interval-valued training data. If the worst point of each training sample to the hyperplane is classified correctly, the whole of the hypercube of each training sample is also classified correctly.

By using basic operations on interval values [15], the model (7) is transformed into the following model:
\[
\begin{align*}
\min_{w,b,\xi_i} \quad & \frac{1}{2} \left\| w \right\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad & y_i \left( w^T m_{\tilde{x}_i} + b \right) - \left\| w \right\|^T s_{\tilde{x}_i} \geq 1 - \xi_i, \quad i = 1, 2, ..., n; \\
\xi_i & \geq 0, \quad i = 1, 2, ..., n;
\end{align*}
\]

where \( |w| = (|w_1|, |w_2|, ..., |w_p|)^T \); \( m_{\tilde{x}_i} = \frac{\tilde{x}_i^U + \tilde{x}_i^L}{2} \) and \( s_{\tilde{x}_i} = \frac{\tilde{x}_i^U - \tilde{x}_i^L}{2} \) are the midpoint and the radius of \( \tilde{x}_i = [\tilde{x}_i^L, \tilde{x}_i^U] \), respectively, and \( \tilde{x}_i^L \) is the lower bound and \( \tilde{x}_i^U \) is the upper bound of \( \tilde{x}_i \).

Now, let’s show the advantage of the robust SVM with respect to the SVM. Consider 10 linear separable interval-valued training samples plotted in Fig. 3. Here, each + (*) symbol shows the midpoint of a sample with the class label +1 (-1). Figs. 4a and 4b show two classifiers obtained for these training samples by using the SVM for \( \varphi(X) = X \) and robust SVM, respectively. We used the midpoint of each interval-valued training sample for the training and test phases of the SVM. As can be seen, the robust SVM has classified samples better than the SVM. The robust SVM has classified the whole of hypercube of samples with the widest symmetric margin.

The main disadvantage of the robust SVM is its inability for nonlinear data classification. Our proposed method uses the robust SVM strategy with fuzzy reasoning for nonlinear data classification.
Fig. 3. 10 linear separable interval-valued samples;
+: midpoint of samples of class +1    *: midpoint of samples of class -1.

Fig. 4. Classification of some interval-valued data by using (a) SVM for $\phi(x) = x$ and $C=1000$, (b) robust SVM and $C=1000$;
+: midpoint of samples of class +1    *: midpoint of samples of class -1.
III. THE PROPOSED METHOD

A. Crisp-input Version

Let \( \{ x_1, x_2, ..., x_{n_p} \} \) \( \subseteq \) class +1 and \( \{ x_{n_p+1}, ..., x_{n_p+n_n} \} \subseteq \) class -1 be \( n_p + n_n \) crisp training samples. Now, consider two training samples \( x_i \) \( \in \) class +1 and \( x_j \) \( \in \) class -1 (See Fig. 5). Based on the hard margin SVM strategy, the optimal hyperplane for these two samples passes through the midpoint of these two samples, namely \( m_{ij} = \frac{(x_i + x_j)}{2} \). In addition, it is orthogonal to the line segment with the endpoints \( x_i \) and \( x_j \) (This hyperplane has widest symmetric margin). Therefore, optimal classifier for these two samples is \( \text{sign}(f_{ij}(X)) \), where \( f_{ij}(X) = w_{ij}^T X + b_{ij} \), \( w_{ij} = x_i - x_j \) is weight vector of the optimal hyperplane, and because the optimal hyperplane passes through \( m_{ij} \), \( b_{ij} = -w_{ij}^T m_{ij} \) is its bias.

![Fig. 5. Optimal hyperplane based-on the hard margin SVM strategy.](image)

We construct the following TS-type rule for each \( x_i \) \( \in \) class +1 and \( x_j \) \( \in \) class -1:

\[ \text{Rule}_{ij}: \text{if } \left( X \text{ is "around } x_i \text{"} \right) \text{ or } \left( X \text{ is "around } x_j \text{"} \right) \text{then } f_{ij}(X) = w_{ij}^T X + b_{ij}; \]
where “around $x_i$” is a fuzzy set. The nearer a member of this fuzzy set to $x_i$, the more its membership value. For example, the membership function of the fuzzy set “around $x_i$” can be considered to be a Gaussian membership function, namely

$$
\mu_{\text{around } x_i}(X) = \exp\left(\frac{\|X - x_i\|^2}{2\sigma^2}\right). \tag{9}
$$

The decision function for the test sample $X$ is given by $\text{sign}(f(X))$, where based-on take-winner-all response,

$$
f(X) = f_{hl}(X), \tag{10}
$$

where $h,l = \arg \max_{i=1,\ldots,n_p} \left\{ \mu_{\text{around } x_i}(X) + \mu_{\text{around } x_j}(X) \right\}$.

**Reducing the Testing Time of Crisp-input Version**

Eq. (10) obtained by using $n_p \times n_p$ rules. This causes to have a high test time. In continue, we show that the time complexity of Eq. (10) can be reduced substantially. We have

$$
\max_{i=1,\ldots,n_p} \left\{ \mu_{\text{around } x_i}(X) + \mu_{\text{around } x_j}(X) \right\} = \\
\max_{k=1,\ldots,n_p} \left\{ \mu_{\text{around } x_k}(X) \right\} + \max_{k=n_p+1,\ldots,n_p+n_n} \left\{ \mu_{\text{around } x_k}(X) \right\}
$$

Therefore, Eq. (10) can be restated as follows:

$$
f(X) = f_{hl}(X), \tag{11}
$$

where
\[ h = \operatorname{arg\,max}_{k=1, \ldots, n_p} \left\{ \mu_{\text{around } x_k}(X) \right\}, \]

\[ l = \operatorname{arg\,max}_{k=n_p+1, \ldots, n_p+n_n} \left\{ \mu_{\text{around } x_k}(X) \right\}. \]

Clearly, obtaining class label of the test sample \( X \) is less time-consuming by using Eq. (11) than by employing Eq. (10). Indeed, the time complexity order of Eq. (11) is equal to the time complexity of an especial fuzzy system obtained by using the following \( n_p + n_n \) rules:

\[
\text{Rule}_k : \text{if} \ (X \ \text{is } \text{around } x_k) \ \text{then} \ f_k(X) = f_{l_}\ (X), \ k = 1, 2, \ldots, n_p + n_n.
\]

**Training Error of the Crisp-input Version**

From Eq. (9), for each \( x_i \in \text{class } +1, \ \mu_{\text{around } x_i}(x_i) = 1 \) and for each \( k \neq t, \ \mu_{\text{around } x_i}(x_i) < 1. \)

Therefore, \( \operatorname{arg\,max}_{k=1, \ldots, n_p} \left\{ \mu_{\text{around } x_i}(x_i) \right\} = t. \) Thus, from Eq. (11), \( f(x_i) = f_{l_}\ (x_i), \) where

\[ l = \operatorname{arg\,max}_{k=n_p+1, \ldots, n_p+n_n} \left\{ \mu_{\text{around } x_i}(x_i) \right\}. \]

Since \( f_{l_}\ () \) classifies \( x_i \) and \( x_i \) correctly, \( \text{sign}(f(X)) = \text{sign}(f_{l_}\ (X)) = +1. \)

In the same way, for each \( x_i \in \text{class } -1, \ \text{sign}(f(X)) = \text{sign}(f_{l_}\ (X)) = -1. \) Thus, training error of the proposed method is zero.

**Margin of Classifier**

As explained earlier, for each \( x_i \in \text{class } +1, \ f(x_i) = f_{l_}\ (x_i), \)

\[ l = \operatorname{arg\,max}_{k=n_p+1, \ldots, n_p+n_n} \left\{ \mu_{\text{around } x_i}(x_i) \right\}. \]

Moreover, \( f_{l_}\ () \) classifies \( x_i \in \text{class } +1 \) and \( x_i \in \text{class } -1 \) with the widest symmetric margin. Therefore, \( f(x_i) \) classifies \( x_i \in \text{class } +1 \) and the nearest member of
class -1 with the widest symmetric margin. As stated earlier, this property can decrease the test error of classifier if both training and test samples are from identical distribution probability.

**Rule Reduction for the Crisp-input Version**

Consider two training samples \( x_i \) and \( x_j \) with identical class label, +1 (or -1), whose features are crisp or in general interval-valued, namely \( x_k = [x_{i_k}^L, x_{i_k}^U] \) and \( x_k = [x_{j_k}^L, x_{j_k}^U] \) for \( k = 1, 2, ..., p \), where \( p \) is training sample dimension. We merge these two training samples with each other and replace them with a new interval-valued sample \( x_i \), where

\[
x_i = \left[ \min\left(x_{i_k}^L, x_{j_k}^L\right), \max\left(x_{i_k}^U, x_{j_k}^U\right) \right]
\]

for \( k = 1, 2, ..., p \), if the intersection of the hypercube of this new interval-valued sample and the hypercube of each \( x_i \in \text{class} -1 \) (or +1) is empty. This procedure must be repeated for each couple of samples of class +1 (or -1).

For example, consider 10 samples plotted in Fig. 6a. Here, the samples of class +1 and -1 are shown by + and *, respectively. We have 10 rules, if no rule reduction is performed. If the mentioned rule reduction procedure is performed, two interval-valued samples and one crisp sample (See Fig. 6b) are obtained. Now, we can use interval-valued-input version of our proposed method, which is explained in the following sub-section to construct some rules for data classification. Because the number of rules which is obtained by using the interval-valued-input version of our proposed method is equal to the number of interval-valued training samples, the number of rules is reduced to 3.
Fig. 6. Reduction of (a) 10 crisp samples to (b) 3 samples (two interval-valued samples and one crisp sample).

+: samples of class +1  *: samples of class -1.

B. Interval-valued-input Version

Let \( \{ \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n_p} \} \in \text{class } +1 \) and \( \{ \tilde{x}_{n_p+1}, \tilde{x}_{n_p+2}, \ldots, \tilde{x}_{n_p+n_n} \} \in \text{class } -1 \) be \( n_p + n_n \) training samples whose feature are interval-valued, namely \( \tilde{x}_{ij} = [\tilde{x}_{ij}^L, \tilde{x}_{ij}^U] \) \( i = 1, 2, \ldots, n_p + n_n; j = 1, 2, \ldots, p \), where \( p \) is training sample dimension. Now, consider two interval-valued training samples \( \tilde{x}_i \in \text{class } +1 \) and \( \tilde{x}_j \in \text{class } -1 \) (See Fig. 7). We construct the classifier of these two samples based on the robust SVM strategy. Based on this strategy, the optimal classifier for these two interval-valued samples is a hyperplane with the widest symmetric margin with respect to the worst point of the hypercube of each training sample called \( w_{p_{ij1}} \) and \( w_{p_{ij2}} \), respectively (See Fig. 7). In the next sub-section, we show that such separating hyperplane can be obtained by using only some “if-then”s. The optimal classifier is \( \text{sign}(f_{ij}(X)) \), where \( f_{ij}(X) = w_{ij}^T X + b_{ij} \), and \( w_{ij} \) and \( b_{ij} \) are the weight vector and bias of the optimal hyperplane obtained by using the robust SVM strategy.
Fig. 7. Optimal hyperplane based-on the robust SVM strategy for \( C=\infty \);

Black circles show the worst points of interval-valued samples.

We construct following TS-type fuzzy if-then rule for each \( \tilde{x}_i \in \text{class } +1 \) and \( \tilde{x}_j \in \text{class } -1 \):

\[
\text{Rule}_{ij} : \text{if } \left( X \text{ is "around } \tilde{x}_i \text{"} \right) \text{ or } \left( X \text{ is "around } \tilde{x}_j \text{"} \right) \text{ then } f_{ij}(X) = w_{ij}^T X + b_{ij},
\]

\[
i = 1, 2, \ldots, n_p; \quad j = 1, 2, \ldots, n_a;
\]

where “around \( \tilde{x}_i \)” is a fuzzy set. Nearer a member of this fuzzy set to \( \tilde{x}_i \), the more is assigned to its membership value. Moreover, the membership value of this fuzzy set for \( \tilde{x}_i^L \leq X \leq \tilde{x}_i^U \) must be 1. For example, the following pseudo-trapezoid function can be used as the membership function of the fuzzy set “around \( \tilde{x}_i \)”:

\[
\mu_{\text{around } \tilde{x}_i}(X) = \prod_{j=1}^{p} \max \left\{ \exp \left( \frac{X_j - \tilde{x}_{ij}^L}{2\sigma^2} \right), \exp \left( \frac{\|X_j - \tilde{x}_{ij}^U\|^2}{2\sigma^2} \right), \text{isBetween}(X_j, \tilde{x}_{ij}^L, \tilde{x}_{ij}^U) \right\},
\]

where

\[
(12)
\]
isBetween \left(X_j, \tilde{x}_j^L, \tilde{x}_j^U \right) = \begin{cases} 1 & \tilde{x}_j^L \leq X_j \leq \tilde{x}_j^U, \\ 0 & \text{otherwise}. \end{cases}

Fig. 8 shows a 1-dimensional case of this membership function. The decision function for the test sample \( X \) is given by \( \text{sign}(f(X)) \), where by using the take-winner-all response,

\[
f(X) = f_{hl}(X),
\]

where \( h, l = \arg \max_{i = 1, \ldots, n_p} \left\{ \mu_{\text{around}}(X) + \mu_{\text{around}}(X) \right\} \).

Reducing the Test Time of the Interval-valued-input Version

As seen, the \( f(X) \) in Eq. (13) has been obtained by using \( n_p \times n_a \) rules. This causes to obtain a high test time. Again, similar to what was explained in the previous sub-section, i.e. sub-section “Reducing the Test Time of the Crisp-input Version”, Eq. (13) can be replaced by

\[
f(X) = f_{hl}(X),
\]
where \( h = \arg \max_{k=1,\ldots,n_p} \{ \mu_{\text{around } \tilde{x}_t}(X) \} \), and \( l = \arg \max_{k=n_p+1,\ldots,n_p+n_n} \{ \mu_{\text{around } \tilde{x}_t}(X) \} \).

Clearly, obtaining class label of the test sample \( X \) is less time-consuming by using Eq. (14) than by employing Eq. (13). Indeed, the time complexity order of Eq. (14) is equal to an especial fuzzy system with the following \( n_p + n_n \) rules:

\[
\text{Rule}_k : \text{if } (X \text{ is "around } \tilde{x}_t\text{"}) \text{ then } f_k(X) = f_{tl}(X), \quad k = 1, 2, \ldots, n_p + n_n;
\]

**Training Error of the Interval-valued-input Version**

Suppose that interval-valued data have no overlapping region. Then, from Eq. (12), for each \( X \) in an interval-valued training data of class \(+1\), i.e. \( \tilde{x}_t \), \( \mu_{\text{around } \tilde{x}_t}(X) = 1 \) and for each \( k \neq t \), \( \mu_{\text{around } \tilde{x}_t}(X) < 1 \). Therefore, \( \arg \max_{k=1,\ldots,n_p} \{ \mu_{\text{around } \tilde{x}_t}(X) \} = t \). Thus, from Eq. (14), \( f(X) = f_{tl}(X) \), where \( l = \arg \max_{k=n_p+1,\ldots,n_p+n_n} \{ \mu_{\text{around } \tilde{x}_t}(X) \} \). Since \( f_{tl}(\cdot) \) classifies the whole of \( \tilde{x}_t \), correctly, \( \text{sign}(f(X)) = \text{sign}(f_{tl}(X)) = +1 \).

In the similar way, for each \( X \) in an interval-valued training data of class \(-1\), i.e. \( \tilde{x}_t \), \( \text{sign}(f(X)) = \text{sign}(f_{tl}(X)) = -1 \). Therefore, each \( X \) in an interval-valued training data is classified, correctly. Thus, training error of the proposed method is zero when interval-valued training data have no overlapping region.

**Margin of Classifier**

As explained earlier, for each \( X \) in an interval-valued training data of class \(+1\), i.e. \( \tilde{x}_t \), \( f(X) = f_{tl}(X) \), where \( l = \arg \max_{k=n_p+1,\ldots,n_p+n_n} \{ \mu_{\text{around } \tilde{x}_t}(X) \} \). Moreover, \( f_{tl}(\cdot) \) classifies the worst point
of \( \tilde{x}_i \) and the worst point of \( \tilde{x}_i \in \text{class } -1 \) with the widest symmetric margin. Therefore, \( f(\cdot) \) classifies the worst point of \( \tilde{x}_i \) and the nearest member of class \(-1\) with the widest symmetric margin. As stated earlier, this property can decrease the test error of classifier if both training and test samples are from identical distribution probability.

**Rule Reduction for the Interval-valued-input Version**

The rule reduction approach (RR) explained earlier for the crisp-input version of our proposed method can be used also for the interval-valued-input version of our proposed method to decrease the test time. For example, consider 10 interval valued samples plotted in Fig. 9a. The midpoint of each sample of class \( +1 \) (-1) has been shown by \( + \) (*) here. We have 10 rules, if no RR is performed. If the mentioned RR procedure is performed, we obtain 5 interval-valued samples (See Fig. 9b). Because the number of rules which is obtained by using the interval-valued-input version of our proposed method is equal to the number of interval-valued training samples, the number of rules is reduced to 5.

Indeed, when two interval-valued samples of class \( +1 \) (-1) are merged to construct a bigger interval-valued sample, the region between those two interval-valued data is labeled to \( +1 \) (-1). However, we are uncertain about the class label of this region. Therefore, the RR can increase the test error. In other words, there is a trade of between the number of rules (which relates to the test time) and misclassification rate.
Fig. 9. Reduction of (a) 10 interval-valued samples to (b) 5 interval-valued samples.

+: samples of class +1  *: samples of class -1.

**Obtaining Optimal Classifier for Two Interval-valued Samples**

Let \(\tilde{x}_i \in \text{class } +1\) and \(\tilde{x}_j \in \text{class } -1\) be two interval-valued training samples. As stated, based on the robust SVM strategy, optimal hyperplane for classification of these two interval-valued training samples is a hyperplane with the widest symmetric margin with respect to the worst points of these two interval-valued training samples. Therefore, it must pass through the midpoint of the two worst points and also must be orthogonal to the line segment between these two worst points. In other words, when two interval-valued training samples are completely separable, the optimal separating hyperplane passes through the midpoint of two nearest points between the hypercubes of the interval-valued samples, but when two training samples are not completely separable, the hyperplane must be as near as possible to the farthest misclassified point of the hypercube of each of two training samples. In this sub-section, we show that such a hyperplane can be obtained by using some “if-then”s.
If there exists at least one $k$ in $\{1, 2, \ldots, p\}$ such that $\tilde{x}_{jk}^U < \tilde{x}_{ik}^L$ or $\tilde{x}_{jk}^L > \tilde{x}_{ik}^U$, $\tilde{x}_i$ and $\tilde{x}_j$ are completely separable (See Fig. 10), else $\tilde{x}_i$ and $\tilde{x}_j$ are overlapping (See Fig. 12). As said, if $\tilde{x}_i$ and $\tilde{x}_j$ are completely separable, the optimal hyperplane passes through the midpoint of two nearest points between the hypercubes of the interval-valued samples called $m = (m_1, m_2, \ldots, m_p)^T$. These two nearest points, i.e. $wp1 = (wp1_1, wp1_2, \ldots, wp1_p)^T$ and $wp2 = (wp2_1, wp2_2, \ldots, wp2_p)^T$, their midpoint and the weight vector of optimal hyperplane $w = (w_1, w_2, \ldots, w_p)^T$ can be obtained as follows (See also Fig. 10):

**Algorithm 1.**

/* Suppose there exists at least one $k$ in $\{1, 2, \ldots, p\}$ such that $\tilde{x}_{jk}^U < \tilde{x}_{ik}^L$ or $\tilde{x}_{jk}^L > \tilde{x}_{ik}^U$, namely $\tilde{x}_i$ and $\tilde{x}_j$ are completely separable, such as those plotted in Fig. 10. Then, compute $w$, $m$, $wp1$ and $wp2$ dimension by dimension as follows:* /

for $k = 1, 2, \ldots, p$ do 

if $|\tilde{x}_{jk}^U - \tilde{x}_{ik}^L| < |\tilde{x}_{jk}^L - \tilde{x}_{ik}^U|$ then /* if the whole of $\tilde{x}_{ik}$ is at right side of $\tilde{x}_{jk}$ (We supposed that $\tilde{x}_i$ and $\tilde{x}_j$ are completely separable) */

// compute $k$-th dimension of $w$, $m$, $wp1$ and $wp2$.

$$w_k = \tilde{x}_{ik}^L - \tilde{x}_{jk}^L, \quad m_k = \frac{\tilde{x}_{ik}^L + \tilde{x}_{jk}^U}{2},$$

$$wp1_k = \tilde{x}_{ik}^L, \quad wp2_k = \tilde{x}_{jk}^U.$$ 

else /* if the whole of $\tilde{x}_{jk}$ is at right side of $\tilde{x}_{ik}$ */


// compute k-th dimension of \( w, m, wp_1 \) and \( wp_2 \).

\[
\begin{align*}
w_k &= \tilde{x}_{ik}^U - \tilde{x}_{jk}^L, \\
m_k &= \frac{\tilde{x}_{ik}^U + \tilde{x}_{jk}^L}{2}, \\
wp_{1k} &= \tilde{x}_{ik}^U, \\
wp_{2k} &= \tilde{x}_{jk}^L.
\end{align*}
\]

Fig. 10. Optimal hyperplane and the worst points of two completely separable interval-valued samples \( \tilde{x}_i \in \text{class } +1 \) and \( \tilde{x}_j \in \text{class } -1 \).

The mentioned procedure fails when some dimensions of \( \tilde{x}_i \) and \( \tilde{x}_j \) are overlapped (See Fig. 11). Therefore, we revise the mentioned procedure as follows:

**Algorithm 2.**

/* Again, suppose there exists at-least one \( k \) in \( \{1, 2, \ldots, p\} \) such that \( \tilde{x}_{jk}^U < \tilde{x}_{ik}^L \) or \( \tilde{x}_{jk}^L > \tilde{x}_{ik}^U \), namely \( \tilde{x}_i \) and \( \tilde{x}_j \) are completely separable. */

for \( k = 1, 2, \ldots, p \) do
if $\tilde{x}_{ik}^L \leq \tilde{x}_{jk}^U$ and $\tilde{x}_{ik}^U \geq \tilde{x}_{jk}^U$ then / if $\tilde{x}_{ik}$ encompasses a part of $\tilde{x}_{jk}$, and $\tilde{x}_{jk}$ is at right side of $\tilde{x}_{ik} 
$ // compute $k$-th dimension of $w$, $m$, $wp1$ and $wp2$. 

$$w_k = 0, \quad m_k = \tilde{x}_{ik}^U$$ 

$wp1_k = \tilde{x}_{ik}^U, \quad wp2_k = \tilde{x}_{ik}^U$.

} 

else if $\tilde{x}_{ik}^L \leq \tilde{x}_{jk}^U$ and $\tilde{x}_{ik}^U \geq \tilde{x}_{jk}^U$ then / if $\tilde{x}_{ik}$ encompasses a part of $\tilde{x}_{jk}$, and $\tilde{x}_{ik}$ is at right side of $\tilde{x}_{jk} 
$ // compute $k$-th dimension of $w$, $m$, $wp1$ and $wp2$. 

$$w_k = 0, \quad m_k = \tilde{x}_{jk}^U$$ 

$wp1_k = \tilde{x}_{jk}^U, \quad wp2_k = \tilde{x}_{jk}^U$.

} 

else if $|\tilde{x}_{ik}^U - \tilde{x}_{ik}^L| < |\tilde{x}_{jk}^L - \tilde{x}_{jk}^U|$ then / if the whole of $\tilde{x}_{ik}$ is at right side of $\tilde{x}_{jk}$ (We supposed that $\tilde{X}_i$ and $\tilde{X}_j$ are completely separable) 

// compute $k$-th dimension of $w$, $m$, $wp1$ and $wp2$. 

$$w_k = \tilde{x}_{ik}^L - \tilde{x}_{jk}^U,$$ 

$m_k = \frac{\tilde{x}_{ik}^L + \tilde{x}_{jk}^U}{2},$ 

$wp1_k = \tilde{x}_{ik}^L, \quad wp2_k = \tilde{x}_{jk}^U$.

} 

else / the whole of $\tilde{x}_{jk}$ is at right side of $\tilde{x}_{ik}$.

// compute $k$-th dimension of $w$, $m$, $wp1$ and $wp2$. 

$$w_k = \tilde{x}_{ik}^U - \tilde{x}_{jk}^L,$$ 

$m_k = \frac{\tilde{x}_{ik}^U + \tilde{x}_{jk}^L}{2},$
If \( \tilde{x}_i \) and \( \tilde{x}_j \) are not completely separable (Fig. 12), the optimal hyperplane must pass through the midpoint of the farthest misclassified point of the hypercubes of interval-valued samples. Therefore, we must find two nearest points between \( \tilde{x}_i \) and \( \tilde{x}_j \), namely \( wp1 \) and \( wp2 \), and then the weight vector \( w \) must be selected such that these two nearest points are misclassified. We use the following procedure to obtain these two nearest points, their midpoint and the weight vector.

**Algorithm 2 (continue).**

// Suppose \( \tilde{x}_i \) and \( \tilde{x}_j \) are not completely separable such as those plotted in Fig. 12.

Suppose \( \min \left( \left\| \tilde{x}_{ik}^L - \tilde{x}_{id}^L \right\|, \left\| \tilde{x}_{id}^L - \tilde{x}_{ik}^U \right\| \right) \leq \min \left( \left\| \tilde{x}_{jk}^U - \tilde{x}_{ik}^U \right\|, \left\| \tilde{x}_{ik}^U - \tilde{x}_{jk}^L \right\| \right) \) for \( k = 1, 2, \ldots, p \). Let \( d \)-th dimension of \( \tilde{x}_i \) and \( \tilde{x}_j \) have the least amount of overlapping. */
for $k = 1, 2, \ldots, d - 1, d + 1, \ldots, p$ do |

/* compute $k$-th dimension of $w$, $m$, $wp1$ and $wp2$, where $k = 1, 2, \ldots, d - 1, d + 1, \ldots, p$. */

\[ w_k = 0; \]

if $\tilde{x}_{ik}^L \leq \tilde{x}_{ik}^U$ and $\tilde{x}_{ik}^U \geq \tilde{x}_{ik}^U$ then |

/* If $\tilde{x}_{ik}$ encompasses a part of $\tilde{x}_{ik}$, and $\tilde{x}_{ik}$ is at right side of $\tilde{x}_{ik}$ */

\[ m_k = \tilde{x}_{ik}^U, \ wp1_k = \tilde{x}_{ik}^U, \ wp2_k = \tilde{x}_{ik}^U. \]

} else if $\tilde{x}_{ik}^L \leq \tilde{x}_{ik}^U$ and $\tilde{x}_{ik}^U \geq \tilde{x}_{ik}^U$ then |

/* If $\tilde{x}_{ik}$ encompasses a part of $\tilde{x}_{ik}$, and $\tilde{x}_{ik}$ is at right side of $\tilde{x}_{ik}$ */

\[ m_k = \tilde{x}_{ik}^U, \ wp1_k = \tilde{x}_{ik}^U, \ wp2_k = \tilde{x}_{ik}^U. \]

} |

// compute $d$-th dimension of $w$, $m$, $wp1$ and $wp2$.

\[ w_d = \left( \frac{\tilde{x}_{jd}^L + \tilde{x}_{jd}^U}{2} \right) - \left( \frac{\tilde{x}_{jd}^L + \tilde{x}_{jd}^U}{2} \right). \]

if $|\tilde{x}_{jd}^L - \tilde{x}_{jd}^U| < |\tilde{x}_{jd}^U - \tilde{x}_{jd}^L|$ then |

/* If $\tilde{x}_{jd}$ is at right side of $\tilde{x}_{jd}$ */

\[ m_d = \left( \frac{\tilde{x}_{jd}^L + \tilde{x}_{jd}^U}{2} \right), \ wp1_d = \tilde{x}_{jd}^U, \ wp2_d = \tilde{x}_{jd}^L, \]

else \[ m_d = \left( \frac{\tilde{x}_{jd}^L + \tilde{x}_{jd}^U}{2} \right), \ wp1_d = \tilde{x}_{jd}^L, \ wp2_d = \tilde{x}_{jd}^U. \]
Fig. 12. Optimal hyperplane and the worst points of two partially overlapping interval-valued samples $\tilde{x}_i \in \text{class } +1$ and $\tilde{x}_j \in \text{class } -1$.

IV. Time Complexity

We have $n$ training samples. These training samples can be reduced to $N << n$ interval-valued training samples by using the rule reduction approach (RR) with the time complexity of $O(n^2)$. Then, the procedure explained in the previous sub-section is called for each couple of interval-valued training samples, namely $N^2$ times. Therefore, the time complexity of our proposed method is $O(n^2 + N^2)$.

The time complexity of subtractive clustering which is used in SCLSE is $O(n^2 + nL)$ [16], where $L$ is the number of clusters. Therefore, the time complexity of our proposed method is less than SCLSE when $N^2 < nL$.

The time complexity of SVM based methods, namely PDFC, SOTFN and BSVM-FC, depends on the algorithm which is used to solve its quadratic model. In this paper, we use “quadprog” function of MATLAB software. The time complexity of this function is about $O(n^{3.2})$ [17]. Therefore, the time complexity of our proposed method is better than PDFC (or SVM), SOTFN and BSVM-FC.
The ISVM-FC uses the idea of learning fuzzy rule parameters through an incremental SVM instead of batch SVM. In other words, the ISVM-FC splits the training set to \( S \) sub-training set. Then, it uses the BSVM-FC to classify the first sub-training set. It then constructs a new training set by (a) some training samples of the previous sub-training sets which are near to the boundary of the obtained classifier and (b) the next sub-training set, and then it uses again the BSVM-FC to classify the new sub-training set. This procedure is continued for every remaining sub-training set. Therefore, the time complexity of ISVM-FC depends on the number of training samples which are near to the boundary of the obtained classifier at each step. If only two training samples of previous sub-training sets are near to the boundary of the obtained classifier at each step, the time complexity of ISVM-FC becomes \( O\left( (S + 2)^{1.2} \times \frac{n}{2} \right) \). The time complexity of ISVM-FC becomes \( O\left( S^{3.2} + (2 \times S)^{3.2} + (3 \times S)^{3.2} + \ldots + n^{3.2} \right) \), if all training samples of each sub-training set are near the classifier at each step. The best-case time complexity of ISVM-FC is better than BSVM-FC and its worst-case time complexity is worse than BSVM-FC and also is worse than our proposed method. More precise comparison of the best-case time complexity of ISVM-FC and our proposed method can be done only when \( n, S \) and \( N \) are known. Our experimental results in section 6 show that the time complexity of our proposed method is better than the time complexity of ISVM-FC in average for some real datasets. Section 6 also compares empirically misclassification rate and the number of rules (which relates to the test time) of our proposed method by those of the other classification methods.
V. NUMERICAL EXAMPLE

We used samples of Fig. 13 as training set and obtained each time a classifier by using the PDFC (or SVM), SCLSE and the interval-valued-input version of our proposed method (with RR and without RR). We used the midpoint of each interval-valued training sample when using the PDFC (or SVM) and SCLSE. However, all of these classification methods could classify the midpoint of training samples correctly by selecting proper value(s) for their parameter(s), but none of them, excepting our proposed method, could classify the whole of hypercube of samples correctly. Fig. 13 shows classifiers obtained by using the mentioned classification methods for optimal value of their parameter(s). We obtained the optimal value of parameter(s) of each classification method by using the grid search method. We say that a parameter value of a classification method is optimal if the most points of boxes of interval-valued training data are classified correctly.

Meanwhile, the number of rules obtained by using our proposed method (without RR) is 10 rules. When RR is used, the number of rules is reduced to 8. Table I also shows the number of rules obtained by using the two other classification methods for the optimal value(s) of their parameter(s).

<table>
<thead>
<tr>
<th>Classification method</th>
<th># of rules / # of SVs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDFC[1] (or SVM[6]) for ( C = 1000 ) and ( \sigma = 1 )</td>
<td>9</td>
</tr>
<tr>
<td>SCLSE[12] for ( R = 0.1 )</td>
<td>4</td>
</tr>
<tr>
<td>Our proposed method (without RR), for ( \sigma = 1 )</td>
<td>10</td>
</tr>
<tr>
<td>Our proposed method (with RR), for ( \sigma = 1 )</td>
<td>8</td>
</tr>
</tbody>
</table>

TABLE I.
THE NUMBER OF RULES (OR THE NUMBER OF SVS) OBTAINED BY USING DIFFERENT CLASSIFICATION METHODS.
Fig. 13. Classification of some interval-valued data by using (a) PDFC (or SVM) for C=1000 and σ=10 (b) SCLSE for R=1, (c) our proposed method (without RR) for σ=10 (d) our proposed method (with RR) for σ=10.

+: midpoint of samples of class +1    *: midpoint of samples of class -1.

VI. APPLICATION

In this section, we compare our proposed method with the mentioned classification methods by using 3 interval-valued datasets and 17 datasets from the UCI repository [18] listed in Tables II
and III, respectively. Before the beginning of training phase of each classification method, we normalize each dataset as follows:

\[ x = \frac{x - \mu}{\sigma}, \]

where \( x \) is a sample feature, and \( \mu \) and \( \sigma \) are the mean and standard deviation of that feature (the mean and standard deviation of the midpoint of that feature for interval-valued datasets).

### Table II.
**Properties of some datasets of UCI repository**

<table>
<thead>
<tr>
<th>#</th>
<th>Dataset</th>
<th># of instances</th>
<th># of features</th>
<th># of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>diagnosis1</td>
<td>120</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>diagnosis2</td>
<td>120</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Hepatitis</td>
<td>155</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Tae</td>
<td>151</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>lung-cancer</td>
<td>32</td>
<td>56</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>heart-cleveland</td>
<td>303</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>Zoo</td>
<td>101</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>breast-tissue</td>
<td>106</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>audiology</td>
<td>200</td>
<td>60</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>Wpbc</td>
<td>198</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>Lenes</td>
<td>24</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>post-operative</td>
<td>90</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>Sonar</td>
<td>208</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>pima</td>
<td>768</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table III.
**Properties of some interval-valued datasets**

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of instances</th>
<th># of features</th>
<th># of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car [19]</td>
<td>33</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Fish [20]</td>
<td>12</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Temperature [21, 22]</td>
<td>37</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

Some of the datasets listed in Tables II and III are multi-class, namely have more than two classes. Solving a multi-class classification problem (see [14, 23, 24]) is, in general, a more difficult task than solving a two-class classification problem. Different strategies have been proposed. Most of these suggest transforming the multi-class problem into a series of two-class
problems, i.e. one-versus-one, one-versus-rest and Decision Directed Acyclic Graphs techniques (see e.g., [25-30]). In this paper, we employ the one-versus-rest technique. Meanwhile, in order to measure the misclassification rate of each classification method, we employ 10-fold cross validation [31].

Finally, we perform each experiment for different values of the parameter(s) of each classification model. These parameters are: penalty term $C$ in linear robust SVM, penalty term $C$ and standard deviation of Gaussian kernel function $\sigma$ in PDFC (or SVM), penalty term $C$ and $\mu_{th}$ in SOTFN-SV, BSVM-FC and ISVM-FC, cluster radius $R$ in SCLSE, and the standard deviation of Gaussian membership function $\sigma$ in our proposed method. We set each time the cluster radius $R$ and standard deviation of Gaussian kernel function $\sigma$ to $\{0.01, 0.02, \ldots, 0.99\}$, the penalty term $C$ to $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, the $\mu_{th}$ to $\{0.01, 0.02, \ldots, 0.99\}$, and the standard deviation of Gaussian membership function $\sigma$ to $\{0.01, 0.02, \ldots, 0.99\}$. All simulations are conducted on a PC with 3GHz Intel central processing unit (CPU) and 1GB RAM, running the Windows XP operating system, with programs compiled by MATLAB.

A. UCI Datasets

Tables IV-VI show the means of misclassification rate, training time and the number of rule obtained by using the mentioned classification methods, respectively. As shown, the mean of misclassification rate of the PDFC (or SVM) is the best; the mean of misclassification rate of our proposed method is better than, or equal to, the mean of misclassification rate of the PDFC (or SVM) and BSVM-FC for 10 datasets; is better than, or equal to, the mean of misclassification rate of the ISVM-FC for 14 datasets; is better than, or equal to, the mean of misclassification rate

---

$\mu_{th}$ is a parameter of the clustering preprocess of SOTFN-SV. A higher number of cluster is generated when $\mu_{th}$ is larger.
of the SOTFN-SV for 15 datasets; and is better than, or equal to, the mean of misclassification rate of SCLSE approach for 12 datasets. The mean of training time of our proposed method is much better than the PDFC (or SVM), ISVM-FC, BSVM-FC and SOTFN-SV for all datasets and is better than SCLSE for 14 datasets. The mean of the number of rules obtained by using our proposed method is better than the PDFC (or SVM) for all datasets; is better than SOTFN-SV and SCLSE for 14 datasets; is better than the ISVM-FC for 10 datasets; and is better than the BSVM-FC for 12 datasets.

### Table IV.
The mean and standard deviation of misclassification rate.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>diagnosis1</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>diagnosis2</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td>0.02±0.01</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>hepatitis</td>
<td>0.12±0.02</td>
<td>0.10±0.03</td>
<td>0.13±0.03</td>
<td>0.13±0.03</td>
<td>0.10±0.05</td>
<td>0.07±0.01</td>
</tr>
<tr>
<td>4</td>
<td>tae</td>
<td>0.35±0.12</td>
<td>0.43±0.15</td>
<td>0.38±0.13</td>
<td>0.42±0.14</td>
<td>0.33±0.11</td>
<td>0.34±0.12</td>
</tr>
<tr>
<td>5</td>
<td>lung-cancer</td>
<td>0.38±0.15</td>
<td>0.35±0.14</td>
<td>0.35±0.15</td>
<td>0.38±0.16</td>
<td>0.38±0.17</td>
<td>0.34±0.13</td>
</tr>
<tr>
<td>6</td>
<td>heart-cleveland</td>
<td>0.39±0.12</td>
<td>0.40±0.16</td>
<td>0.40±0.19</td>
<td>0.39±0.11</td>
<td>0.40±0.17</td>
<td>0.37±0.12</td>
</tr>
<tr>
<td>7</td>
<td>glass</td>
<td>0.30±0.09</td>
<td>0.31±0.13</td>
<td>0.27±0.09</td>
<td>0.31±0.16</td>
<td>0.29±0.10</td>
<td>0.28±0.11</td>
</tr>
<tr>
<td>8</td>
<td>zoo</td>
<td>0.04±0.01</td>
<td>0.06±0.02</td>
<td>0.05±0.02</td>
<td>0.09±0.03</td>
<td>0.05±0.02</td>
<td>0.05±0.02</td>
</tr>
<tr>
<td>9</td>
<td>wine</td>
<td>0.02±0.01</td>
<td>0.04±0.01</td>
<td>0.03±0.01</td>
<td>0.04±0.01</td>
<td>0.02±0.01</td>
<td>0.04±0.01</td>
</tr>
<tr>
<td>10</td>
<td>iris</td>
<td>0.02±0.01</td>
<td>0.04±0.01</td>
<td>0.04±0.01</td>
<td>0.07±0.02</td>
<td>0.03±0.01</td>
<td>0.05±0.01</td>
</tr>
<tr>
<td>11</td>
<td>breast-tissue</td>
<td>0.25±0.10</td>
<td>0.34±0.13</td>
<td>0.32±0.14</td>
<td>0.30±0.13</td>
<td>0.29±0.11</td>
<td>0.29±0.12</td>
</tr>
<tr>
<td>12</td>
<td>audiology</td>
<td>0.19±0.08</td>
<td>0.22±0.09</td>
<td>0.20±0.07</td>
<td>0.24±0.11</td>
<td>0.28±0.10</td>
<td>0.21±0.07</td>
</tr>
<tr>
<td>13</td>
<td>wpbc</td>
<td>0.19±0.07</td>
<td>0.23±0.09</td>
<td>0.20±0.08</td>
<td>0.20±0.10</td>
<td>0.25±0.10</td>
<td>0.23±0.10</td>
</tr>
<tr>
<td>14</td>
<td>lenes</td>
<td>0.10±0.03</td>
<td>0.06±0.03</td>
<td>0.06±0.03</td>
<td>0.10±0.04</td>
<td>0.07±0.02</td>
<td>0.09±0.03</td>
</tr>
<tr>
<td>15</td>
<td>post-operative</td>
<td>0.28±0.08</td>
<td>0.29±0.09</td>
<td>0.28±0.07</td>
<td>0.29±0.09</td>
<td>0.29±0.08</td>
<td>0.30±0.09</td>
</tr>
<tr>
<td>16</td>
<td>sonar</td>
<td>0.10±0.02</td>
<td>0.12±0.03</td>
<td>0.12±0.02</td>
<td>0.13±0.05</td>
<td>0.13±0.03</td>
<td>0.10±0.03</td>
</tr>
<tr>
<td>17</td>
<td>pima</td>
<td>0.23±0.02</td>
<td>0.24±0.04</td>
<td>0.25±0.06</td>
<td>0.25±0.07</td>
<td>0.24±0.05</td>
<td>0.23±0.03</td>
</tr>
</tbody>
</table>

### Table V.
The mean of the number of rules (or the number of SVs).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>diagnosis1</td>
<td>12.12</td>
<td>4.34</td>
<td>3.45</td>
<td>8.34</td>
<td>8.46</td>
<td>4.36</td>
</tr>
<tr>
<td>2</td>
<td>diagnosis2</td>
<td>11.43</td>
<td>2.54</td>
<td>4.23</td>
<td>6.98</td>
<td>9.24</td>
<td>3.45</td>
</tr>
<tr>
<td>3</td>
<td>hepatitis</td>
<td>71.45</td>
<td>13.56</td>
<td>16.67</td>
<td>69.34</td>
<td>114.64</td>
<td>10.63</td>
</tr>
<tr>
<td>4</td>
<td>tae</td>
<td>97.56</td>
<td>20.56</td>
<td>76.25</td>
<td>24.16</td>
<td>25.25</td>
<td>36.63</td>
</tr>
<tr>
<td>5</td>
<td>lung-cancer</td>
<td>29.45</td>
<td>29.56</td>
<td>29.95</td>
<td>29.84</td>
<td>29.56</td>
<td>8.64</td>
</tr>
<tr>
<td>6</td>
<td>heart-cleveland</td>
<td>128.54</td>
<td>11.34</td>
<td>8.36</td>
<td>115.75</td>
<td>272.75</td>
<td>40.64</td>
</tr>
<tr>
<td>7</td>
<td>glass</td>
<td>128.46</td>
<td>25.45</td>
<td>52.63</td>
<td>14.56</td>
<td>160.75</td>
<td>17.64</td>
</tr>
<tr>
<td>8</td>
<td>zoo</td>
<td>33.23</td>
<td>11.89</td>
<td>16.87</td>
<td>55.42</td>
<td>15.96</td>
<td>5.64</td>
</tr>
</tbody>
</table>
have been proposed for crisp data classification. Therefore, we use the midpoint of interval valued datasets are too small. SCLSE proposed method B.

\[
\begin{array}{cccccc}
1 & \text{diagnosis}1 & 1.0222 & 0.6533 & 1.0318 & 1.0100 & 0.0130 & 0.0115 \\
2 & \text{diagnosis}2 & 0.9892 & 0.5303 & 0.8217 & 0.9240 & 0.0136 & 0.0125 \\
3 & \text{hepatitis} & 1.5089 & 0.9068 & 1.9224 & 3.5026 & 0.3817 & 0.0250 \\
4 & \text{tae} & 3.5619 & 2.1096 & 4.1407 & 2.1121 & 0.0281 & 0.0440 \\
5 & \text{lung-cancer} & 0.0392 & 0.2075 & 0.2011 & 0.2594 & 0.1716 & 0.0075 \\
6 & \text{heart-cleveland} & 3.9943 & 2.6129 & 9.2094 & 25.5144 & 2.7230 & 0.1221 \\
7 & \text{glass} & 3.9943 & 2.7113 & 5.4952 & 3.8573 & 0.5472 & 0.0435 \\
8 & \text{zoo} & 0.8242 & 0.4247 & 0.6617 & 1.8541 & 0.0341 & 0.0120 \\
9 & \text{wine} & 1.7367 & 1.5857 & 2.1946 & 2.1481 & 0.0242 & 0.0292 \\
10 & \text{iris} & 1.5093 & 1.1152 & 1.5820 & 1.5990 & 0.0088 & 0.0521 \\
11 & \text{breast-tissue} & 0.4574 & 0.7771 & 0.8120 & 0.8222 & 0.1171 & 0.0111 \\
12 & \text{audiology} & 29.2869 & 2.1781 & 8.5246 & 6.7776 & 2.2675 & 0.0541 \\
13 & \text{wpbc} & 3.7898 & 2.5784 & 3.5877 & 12.0509 & 1.7547 & 0.0753 \\
14 & \text{lenses} & 0.0244 & 0.0794 & 0.0779 & 0.0984 & 0.0142 & 0.0032 \\
15 & \text{post-operative} & 0.4583 & 0.7386 & 0.5946 & 1.5011 & 0.0481 & 0.0231 \\
16 & \text{sonar} & 4.3087 & 8.5576 & 12.1134 & 25.9413 & 2.7379 & 0.1673 \\
17 & \text{pima} & 64.4563 & 48.4857 & 62.7564 & 55.9467 & 16.3765 & 1.6754 \\
\end{array}
\]

\textbf{Table VI. The mean of training time.}

In this sub-section, we use three interval-valued datasets listed in table III to compare our proposed method with the PDFC (or SVM), linear robust SVM, SOTFN-SV, BSVM-FC and SCLSE. We don’t compare our proposed method with ISVM-FC, because these three interval-valued datasets are too small; ISVM-FC is similar to BSVM-FC, and is used for larger datasets.

Interval-valued-input version of our proposed method and the linear robust SVM can be used for interval-valued data classification but the PDFC (or SVM), SOTFN-SV, BSVM-FC and SCLSE have been proposed for crisp data classification. Therefore, we use the midpoint of each training
sample of interval-valued datasets as training set of each of these crisp data classification methods.

Tables VII-IX show means of misclassification rate, training time and the number of rules obtained by different classification methods for ‘car’, ‘fish’ and ‘temperature’ datasets, respectively. Test time of each classification method is related to the number of rules. As shown, mean of training time of our proposed method is much lower than mean of training time of all other classification methods, and mean of its misclassification rate and mean of the number of rules obtained by using our proposed method are at least comparable with the other classification methods.

<table>
<thead>
<tr>
<th>Table VII.</th>
<th>COMPARISON OF DIFFERENT CLASSIFICATION METHODS BY USING THE ‘CAR’ DATASET.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean &amp; Std. of mis. Rate</td>
<td>0.15±0.03</td>
</tr>
<tr>
<td>Mean of training time</td>
<td>0.0451</td>
</tr>
<tr>
<td>Mean of the # of rules</td>
<td>8.34</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table VIII.</th>
<th>COMPARISON OF DIFFERENT CLASSIFICATION METHODS BY USING THE ‘FISH’ DATASET.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean &amp; Std. of mis. Rate</td>
<td>0.16±0.04</td>
</tr>
<tr>
<td>Mean of training time</td>
<td>0.0159</td>
</tr>
<tr>
<td>Mean of the # of rules</td>
<td>8.53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table IX.</th>
<th>COMPARISON OF DIFFERENT CLASSIFICATION METHODS BY USING THE ‘TEMPERATURE’ DATASET.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean &amp; Std. of mis. Rate</td>
<td>0.05±0.01</td>
</tr>
<tr>
<td>Mean of training time</td>
<td>0.0538</td>
</tr>
<tr>
<td>Mean of the # of rules</td>
<td>5.53</td>
</tr>
</tbody>
</table>
VII. CONCLUSION

In each of the traditional SVM-based fuzzy system, an SVM model is solved. Solving the SVM model is time consuming. To overcome this disadvantage, we proposed a rapid method to solve the robust SVM model and used it for tuning the consequent parameters.

The SVM model classifies data with the widest symmetric margin. This property decreases the probability of misclassification of classifier if both training and test samples are from identical probability distribution. But as explained earlier, the mentioned SVM-based models do not maintain the widest symmetric margin of classifier because of the use of rule reduction or decreasing the number of support vectors. As explained, the proposed method classifies each two nearest training samples of two different class labels with the widest symmetric margin. However, our proposed rule reduction procedure does not maintain the widest symmetric margin of classifier, too.

According to experimental results, the test error of the PDFC (or SVM) is the best, and the test error of our proposed method is often lower than the test error of SOTFN-SV, ISVM-FC, BSVM-FC and SCLSE for the mentioned datasets of UCI repository. Therefore, it seems that the classifier of our proposed method has better symmetric margin than the classifiers of the other SVM-based fuzzy systems, i.e. SOTFN-SV, ISVM-FC and BSVM-FC.

The higher test error of our proposed method with respect to the PDFC (or SVM) for some of the datasets may be for the use of rule reduction in our proposed method. As stated, the mentioned rule reduction can increase the test error because it assigns some unlabeled regions to one of the two classes in a two-class classification. In future, we try to add some intelligence to the rule reduction procedure to suppress this disadvantage. For example, we can only merge
those neighbors of the same class which are not farther than a threshold to minify the mentioned unlabeled regions.

As stated, the time complexity of the proposed method for an especial value of its parameter is almost lower than those of the mentioned rule reduction methods. Meanwhile, finding the optimal value of the parameter(s) of a rule extraction or classification method is time-consuming. Our proposed method has only one parameter where as all traditional SVM-based rule reduction methods have more than one parameter.

Finally, according to experimental results, training time and also the number of rules obtained by our proposed method are often better than the PDFC (or SVM), SOTFN-SV, ISVM-FC, BSVM-FC and SCLSE approaches.

REFERENCES


