Constructing L2-SVM Based Fuzzy Classifiers in High-Dimensional Space with Automatic Model Selection and Fuzzy Rule Ranking

Shang-Ming Zhou, *Member, IEEE* and John Q. Gan, *Senior Member, IEEE*

***Abstract*—In this paper, a new scheme for constructing parsimonious fuzzy classifiers is proposed based on the L2-support vector machine (L2-SVM) technique with model selection and feature ranking performed simultaneously in an integrated manner, in which fuzzy rules are optimally generated from data by L2-SVM learning. In order to identify the most influential fuzzy rules induced from the SVM learning, two novel indices for fuzzy rule ranking are proposed and named as α-values and ω-values of fuzzy rules in this paper. The α-values are defined as the Lagrangian multipliers of the L2-SVM and adopted to evaluate the output contribution of fuzzy rules, while the ω-values are developed by considering both the rule base structure and the output contribution of fuzzy rules. As a prototype based classifier, the L2-SVM based fuzzy classifier evades the curse of dimensionality in high-dimensional space in the sense that the number of support vectors, which equals the number of induced fuzzy rules, is not related to the dimensionality. Experimental results on high-dimensional benchmark problems have shown that by using the proposed scheme the most influential fuzzy rules can be effectively induced and selected, and at the same time feature ranking results can also be obtained to construct parsimonious fuzzy classifiers with better generalization performance than the well-known algorithms in literature.**

***Index Terms*—Feature Ranking, Fuzzy Classifier, L2-SVM, Prototype Based Classifier, Rule Induction, Rule Ranking.**

I. Introduction

SVM and kernel based learning systems are a powerful class of algorithms for classification or regression. The advantage of the SVM learning algorithm lies in that based on quadratic programming it leads to parsimonious model structure for regression and classification [1]. In data-driven fuzzy modeling, the commonly used scheme for achieving a parsimonious fuzzy system is to perform rule base reduction by removing redundant rules based on heuristic criteria or selecting relevant variables based on their influence on the output. Although some researchers have made efforts to apply the "kernel tricks" to fuzzy systems for regression and classification [2][3][4], the advantage of the SVM in yielding parsimonious solutions has not been fully exploited in fuzzy systems yet. This is mainly because it is difficult to link the basis functions or membership functions (MFs) used in fuzzy systems to the kernel functions used in the SVM. Chen and Wang [5] indicated that if the MFs associated with the same input variable are generated from location transformation of a reference function [6], then the *if-part* in each fuzzy rule defined as the *t-norm* of every variable's MF is proven to be a Mercer kernel under the condition that the Fourier transform of the reference function is non-negative [7]. Thus, fuzzy classifiers can be constructed based on the SVM technique, leading to a parsimonious model structure. However, one challenging problem has not been addressed in [5] for the SVM based fuzzy classifier, that is, how to select optimal parameters for kernel functions and the regularization parameter in SVM so as to improve the generalization performance.

As a matter of fact, the problem of optimal kernel parameter selection for kernel functions remains open for most kernel machine models [4][8][9][10][11]. Facing so many parameters in the SVM based fuzzy classifier, methods based on exhaustive search become intractable. Recently, Chapelle et al suggested a technique of choosing parameters for SVMs by minimizing radius-margin bound [12]. However, the radius-margin bound only holds in L2-SVM. For the L1-SVM, which was used in [5] for constructing parsimonious fuzzy classifier, the radius-margin bound can not be applied. In order to perform the automatic model selection in SVM based fuzzy classification, this paper proposes a L2-SVM based fuzzy classifier construction method which automatically choose the number of fuzzy rules and identify the important input features at the same time.

It is noteworthy that fuzzy rule selection is an important issue in fuzzy systems. Even though the SVM learning produces sparse support vectors, it is demonstrated in our experiments that there exist redundant or correlated fuzzy rules in the fuzzy classifier initially induced by the L2-SVM learning and that a fuzzy rule selection procedure results in more parsimonious L2-SVM based fuzzy classifiers with better generalization performance. Currently, in traditional

Manuscript received February 8, 2005; revised July 26, 2005.

S. M. Zhou is with the Department of Computer Science, University of Essex, Colchester, CO4 3SQ, UK (phone: +44-(0)1206-874381; fax: +44-(0)1206-872788; e-mail: smzhou@ieee.org).

J. Q. Gan is with the Department of Computer Science, University of Essex, Colchester, CO4 3SQ, UK (e-mail: jqgan@essex.ac.uk).

fuzzy modeling, one strategy for rule ranking and rule subset selection that has received much attention in recent literature [13][14][15][16][17] is based on the singular value decomposition (SVD) of the firing strength matrix (FSM) of fuzzy rules. Specifically speaking, SVD-QR with column pivoting algorithm is applied to the FSM to produce rule ranking information. However, the rule ranking result by the SVD-QR with column pivoting algorithm is heavily dependent on the estimation of an effective rank which is related to the number of near zero singular values. The problem is that there is usually no clear gap between the small singular values and other "large" singular values, and different ranks often produce dramatically different rule ranking results [17]. A method to avoid the estimation of the effective rank is to apply the pivoted QR decomposition directly to the FSM, in which the *R*-values defined as the absolute values of diagonal elements of matrix *R* in QR decomposition tend to track the singular values of the FSM [17][18] and can be used for rule ranking to identify the influential rules. However, when the pivoted QR decomposition algorithm or the SVD-QR with column pivoting algorithm is applied to fuzzy rule ranking, they ignore the effects of the rule consequents. A more effective rule ranking should consider the output contribution of the fuzzy rules [17]. Moreover, as indicated in [17], it is highly expected for a rule ranking method to take into account both the rule base structure and the output contribution of fuzzy rules in order to generate a compact rule base with good generalization performance. To the best of our knowledge, this kind of more reasonable rule ranking scheme (i.e. taking into account both the rule base structure and the output contribution of fuzzy rules) has not been reported in literature yet. Thereupon, this paper is also committed to address this difficulty.

In fact, in the L2-SVM induced fuzzy classifier, the Lagrangian multipliers $\alpha = [\alpha^{(1)}, \cdots, \alpha^{(N)}]^T$ of the SVM (where *N* is the number of training samples) are closely related to the effect of the rule consequents and can be considered as the measures of the output contribution of fuzzy rules. In this paper two novel rule ranking indices named as α-values and ω-values of fuzzy rules are proposed in terms of α. The rule ranking by α-values takes into account the output contribution of induced fuzzy rules but ignores the rule base structure, while the rule ranking by ω-values considers both the rule base structure and the output contribution of fuzzy rules.

The organization of this paper is as follows. Section II describes a new L2-SVM based fuzzy classification algorithm. Two new fuzzy rule ranking indices and a fuzzy rule subset selection procedure are proposed in Section III. Section IV evaluates the performance of the proposed scheme with high-dimensional benchmark problems, followed by discussions about additional advantages of the proposed scheme in Section V. Section VI concludes the paper.

II. The Proposed L2-SVM based Fuzzy Classification System

A. Formulation of the L2-SVM Based Fuzzy Classifier

Consider a fuzzy model with *L* fuzzy rules in the following form:

$$R_i : if\ x_1\ is\ A_i^1\ and \cdots and\ x_n\ is\ A_i^n\ then\ y_i = b_i \quad (1)$$

where *i*=1, 2, …, *L*, x_j and y_i are the input and output variables of the *i*th rule R_i respectively, and A_i^j are the linguistic labels expressed as fuzzy sets with specific semantic meanings of behaviors of the system being modeled, which are characterized by membership functions $A_i^j(x_j)$ generated by expert knowledge or from data, b_i is the consequent parameter of the *i*th rule. In order for the input space to be thoroughly covered by the fuzzy rule "patches", the following auxiliary rule is added into the rule base [5]:

$$R_0 : if\ x_1\ is\ A_0^1\ and \cdots and\ x_n\ is\ A_0^n\ then\ y_0 = b_0 \quad (2)$$

where A_0^j denotes the domain of x_j and $A_0^j(x_j) \equiv 1$, and $b_0 \in \Re$. The overall output of the system is expressed by

$$F(x) = \frac{b_0 + \sum_{i=1}^{L} \tau_i(x) b_i}{1 + \sum_{i=1}^{L} \tau_i(x)} \quad (3)$$

where τ_i is the *firing strength* of the *i*th rule and is usually calculated in terms of an appropriate *T-norm* operator such as the product as follows:

$$\tau_i(x) = \prod_j A_i^j(x_j) \quad (4)$$

Apparently, this is a zero order Takagi-Sugeno (TS) fuzzy system [19], a kind of linguistic model with attractive properties such as the automatic determination of system parameters from data [20]. A binary fuzzy classifier can be defined as follows:

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{L} \tau_i(x) b_i + b_0 \right) \quad (5)$$

In order to apply SVM learning to (5), $\tau_i(x)$ must be a Mercer kernel. Fortunately, as analyzed in [5], if the MFs $A_i^j(x_j)$ are generated from a reference function $a^j(\cdot)$ through location shift [6], i.e., $A_i^j(x_j) = a^j\left(x_j - m_i^j\right)$, and the Fourier transform of the reference function is non-negative, then $\tau_i(x)$ is proved to be a Mercer kernel. There are several reference functions defined in [5] that ensure the

multi-dimensional MF $\tau_i(x)$ to be a Mercer kernel. In this paper, the following reference function is adopted:

$$a^j(r) = e^{-\eta_j r^2} \quad (\eta_j > 0) \tag{6}$$

whose Fourier transform is non-negative, hence $\tau_i(x) = \tau(x, m_i) = \prod_{j=1}^{n} a^j(x_j - m_i^j)$ is a Mercer kernel, where $m_i = \left(m_i^1, \cdots, m_i^n\right)^T$ is called prototype or kennel centre. It should be noted that parameters η_j in the reference function (6) are kernel parameters indicating the importance of input variables, which were manually selected in the modelling scheme used in [5]. However, it is impractical to manually choose different values of η_j for different features in a high-dimensional input space in order to obtain a classification system with good generalization performance. This paper adopts a learning scheme to automatically update parameters η_j.

In order to perform input feature/variable ranking automatically, input variables are scaled by the following modulator function:

$$\hat{x}_j = x_j \theta_j \tag{7}$$

where $\theta_j \in (0, 1)$ indicates the importance of the input variable x_j to the classification task and is defined as

$$\theta_j = 1 - e^{-\varphi_j^2} \tag{8}$$

where $\varphi_j \in \Re$. The above definition of θ_j is to ensure $\theta_j \in (0, 1)$ when φ_j is adjusted by a learning algorithm. Let $\eta_j = \theta_j^2$ in (6), then a SVM based fuzzy classifier can be expressed as

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{L} \tau_\theta(x, m_i) b_i + b_0 \right) \tag{9}$$

where $\tau_\theta(x, m_i) = \prod_{j=1}^{n} a^j(x_j - m_i^j) = \prod_{j=1}^{n} e^{-\theta_j^2 (x_j - m_i^j)^2}$. It can be seen from (1), (4) and (9) that each $\tau_\theta(x, m_i)$ corresponds to a fuzzy rule. To construct a L2-SVM based fuzzy classifier described by (9), the following parameters should be determined: the number of rules L, prototypes m_i, weights b_i, bias b_0, and scaling parameters θ_j.

Given a training dataset $\left\{ x^{(l)}, y^{(l)} \right\}_{l=1}^{N}$, where $y^{(l)} \in \{-1, 1\}$, the L2-SVM learning algorithm seeks the optimal hyperplane with maximal margin by minimizing the following function over $\alpha = [\alpha^{(1)}, \cdots, \alpha^{(N)}]^T$:

$$W(\alpha, \theta) = \sum_{l=1}^{N} \alpha^{(l)} - \frac{1}{2} \sum_{l,k=1}^{N} \alpha^{(l)} \alpha^{(k)} y^{(l)} y^{(k)} \tilde{\tau}_\theta(x^{(l)}, x^{(k)}) \tag{10}$$

under the constraints

$$\sum_{l=1}^{N} \alpha^{(l)} y^{(l)} = 0 \text{ and } 0 \le \alpha^{(l)} \tag{11}$$

where $\tilde{\tau}_\theta(x^{(l)}, x^{(k)})$ is a kernel function used in the L2-SVM and is defined as

$$\tilde{\tau}_\theta(x^{(l)}, x^{(k)}) = \tau_\theta(x^{(l)}, x^{(k)}) + \delta_{lk} / C \tag{12}$$

and $\delta_{lk} = 1$ for $l = k$, and 0 for $l \neq k$, C is a regularization parameter penalizing the training error. By solving the dual optimization problem (10)~(11), one obtains the optimal coefficient vector $\alpha_0 = \left[\alpha_0^{(1)}, \cdots, \alpha_0^{(N)} \right]^T$. There would be many zero coefficients in α_0, and only those samples that correspond to non-zero coefficients will play a role in the determination of model parameter values and are called support vectors. Let L be the number of non-zero coefficients which are denoted as $\tilde{\alpha}_0^{(i)}$. Then the output of the ith fuzzy rule can be calculated as

$$b_i = \tilde{\alpha}_0^{(i)} \tilde{y}^{(i)} \tag{13}$$

where $\tilde{y}^{(i)}$, i=1, 2, …, L, are the class labels of the corresponding support vectors. Hence, the non-linear decision function (9) becomes

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{L} \tau_\theta(x, \tilde{x}^{(i)}) \tilde{\alpha}_0^{(i)} \tilde{y}^{(i)} + b_0 \right) \tag{14}$$

where $\tilde{x}^{(i)}$ represent support vectors which will be set as prototypes m_i in fuzzy rule induction, and the bias term b_0 can be computed as follows,

$$b_0 = \frac{1}{L} \sum_{j=1}^{L} \left(\tilde{y}^{(j)} - \sum_{i=1}^{L} \tilde{\alpha}_0^{(i)} \tilde{y}^{(i)} \tau_\theta(\tilde{x}^{(j)}, \tilde{x}^{(i)}) \right) \tag{15}$$

In the above solution, the values of θ_j and C are assumed known. In [5] these parameter values are chosen manually. This paper automatically identifies the values of θ_j and C from data based on L2-SVM techniques. The following radius-margin bound [1] is adopted in this paper as the objective function:

$$J = \frac{S(\theta)}{\gamma(\theta)} \tag{16}$$

where $S(\theta)$ represents the squared radius of the smallest sphere containing all the training samples in the feature space and $\gamma(\theta)$ denotes the squared margin from the SVM hyperplane to the closest training sample. It was shown [1] that the margin can be expressed as

$\gamma(\theta) = 1/\left(2W(\alpha_0, \theta)\right)$. Therefore, the radius-margin bound becomes

$$J = 2S(\theta) \cdot W(\alpha_0, \theta) \tag{17}$$

On the other hand, the squared radius of the smallest sphere enclosing all the training samples can be estimated by solving the following quadratic programming problem [1]:

$$S(\theta) = \max_{\beta} \left(\sum_{l=1}^{N} \beta^{(l)} \tilde{\tau}_{\theta}(x^{(l)}, x^{(l)}) - \sum_{l,k=1}^{N} \beta^{(l)} \beta^{(k)} \tilde{\tau}_{\theta}(x^{(l)}, x^{(k)}) \right) \tag{18}$$

subject to

$$\sum_{l=1}^{N} \beta^{(l)} = 1 \text{ and } 0 \le \beta^{(l)} \tag{19}$$

Parameters φ_j (θ_j by (8)) and C can be learnt optimally from data in terms of the gradients of J with respect to φ_j and C respectively. Detailed analysis of training L2-SVM can be found in [12].

B. Extraction of Fuzzy Rules after the L2-SVM Learning Process

After the L2-SVM learning process is completed, a parsimonious fuzzy classifier can be induced, in which the fuzzy rules are extracted in the form of (1) based on the decision function of the SVM. Specifically speaking, the induction process is performed as follows:

Step 1. Set the number of fuzzy rules as the number of support vectors;

Step 2. The premise parts of fuzzy rules are evaluated from support vectors and modulator function values: the MFs of the *i*th rule are $A_i^j(x_j) = a^j(x_j - m_i^j)$, where m_i^j is the *j*th element of the *i*th support vector $\tilde{x}^{(i)} \in \{x^{(l)}\}_{l=1}^{N}$ ($i = 1, \cdots, L$). Here, the learned value of φ_j is used to calculate the kernel parameter θ_j in (8).

Step 3. The consequent parts of fuzzy rules are induced from α_0 and class labels: the consequent value of the *i*th rule is $b_i = \tilde{\alpha}_0^{(i)} \tilde{y}^{(i)}$, $i = 1, \cdots, L$, where $\tilde{\alpha}_0^{(i)}$ represent non-zero $\alpha_0^{(l)}$, and $\tilde{y}^{(i)}$ the class labels corresponding to the *L* support vectors.

C. Feature Ranking

After the L2-SVM learning process, the goodness of features can also be identified based on the values of parameters θ_j. It is clear that a larger value of θ_j indicates that feature x_j is more important. In this paper the most appropriate features relevant to the classification task are identified based on a relative ranking index $\tilde{\theta}_j$ defined as follows:

$$\tilde{\theta}_j = \frac{\theta_j}{\max_j \theta_j} \tag{20}$$

D. A Comparison between the Proposed Method and Radial Basis Function Classifier

Radial basis function networks (RBFNs) have been a topic of extensive research with wide applications in machine learning and engineering. The output of a binary RBFN classifier can be computed by the following expression

$$f(x) = \text{sgn}(\sum_{i=1}^{M} \varpi_i \varphi(\| x - c^{(i)} \|) \tag{21}$$

where $\varphi(\| x - c^{(i)} \|)$ are called radial basis functions (RBFs) with prototypes $c^{(i)}$, and ϖ_i are the network weights. From (9) and (21), it can be seen that the proposed classifier and RBFN classifier have a similar decision function for classification, and both RBFN classifiers and SVM based classifiers can be interpreted as fuzzy classifiers. Some researchers actually suggested to treat RBFN as a special case of SVM [21]. However, there are essential differences between RBFN classifiers and SVM based classifiers. Firstly the learning objective functions and the learning algorithms are substantially different. The parameters of a RBFN can be learned via nonlinear optimization using Levenberg–Marquardt method [22][23], evolutionary algorithm [24], EM algorithm [25], or structured nonlinear optimization method [26]. Additionally, the network prototypes are usually determined via other means such as unsupervised clustering algorithms, and the linear weights may then be estimated by the standard least squares solution. Obviously, although this sort of method using least square techniques may give a rough approximation, it cannot yield optimal parameters [26]. Moreover, the number of prototypes in RBFN has to be determined via other means, such as cross validation or cluster validity index. Another interesting approach to constructing RBFNS is to use the orthogonal least squares (OLS) algorithm to identify a parsimonious RBFN by formulating the problem as a linear learning one [27], in which training samples act as candidate RBFN prototypes.

Unlike RBFN, the invention of SVM was driven by underlying statistical learning theory, i.e., following the principle of structural risk minimization that is rooted in VC dimension theory, which makes its derivation even more profound [28]. Vapnik's theory [1] shows that the SVM solution is found by minimizing both the error on the training set (empirical risk) and the complexity of the hypothesis space, expressed in terms of VC-dimension. In this sense, the

decision function found by SVM is a tradeoff between learning error and model complexity. Hence, SVM classifiers usually achieve good generalization performance. Additionally, SVMs have a clear geometrical interpretation and a global minimum of the cost function can be surely found by SVM training, because the parameters of a SVM, including the number of kernel functions, their prototypes, i.e., support vectors, and the linear weights and bias levels, are determined by solving a convex quadratic programming problem with linear inequality and equality constraints. Except for the kernel function parameters, the above mentioned parameters of a SVM are all computed automatically in one model structure. The proposed L2-SVM based fuzzy classifier not only inherits the above properties of SVM, but also learns the kernel function parameters adaptively from data.

III. Fuzzy Rule Ranking and Rule Subset Selection

The parsimony of the L2-SVM based fuzzy classifier hails from the inherent sparse solutions in the SVM, i.e., the support vectors with non-zero Lagrangian multipliers $\tilde{\alpha}_0^{(i)}$. However, these induced fuzzy rules are equally treated in the induced fuzzy classifier without fuzzy rule selection. In this section, a fuzzy rule ranking is produced according to the importance of induced fuzzy rules, aiming to generate a more parsimonious fuzzy classifier based on the most influential fuzzy rules. First the so-called *R*-values of fuzzy rules are briefly introduced. After that, two new indices for fuzzy rule ranking and a fuzzy rule selection procedure are developed.

A. *R*-values of Fuzzy Rules

For an induced fuzzy classifier, its FSM is defined as follows:

$$G = \begin{bmatrix} g_1(1) & \cdots & g_L(1) \\ \vdots & \vdots & \vdots \\ g_1(N) & \cdots & g_L(N) \end{bmatrix}_{N \times L} \tag{22}$$

where

$$g_i(k) = \frac{\tau_i(x^{(k)})}{\sum_{i=1}^{L} \tau_i(x^{(k)})} \tag{23}$$

It can be seen that each column of the matrix *G* corresponds to one fuzzy rule. Therefore, the important fuzzy rules correspond to the columns that are linearly independent of each other. As indicated in [14][15], redundant fuzzy rules (corresponding to linearly dependent or zero-valued columns) are associated with near zero singular values of *G*. As a matter of fact, the smaller are the singular values, the less influential are the associated rules, which is the starting point of the SVD-QR with column pivoting algorithm and the pivoted QR decomposition algorithm that have been applied to fuzzy rule ranking [13][14][15][16][17]. The pivoted QR decomposition algorithm for ranking fuzzy rules is summarized as follows:

1) Calculate the QR decomposition of *G* and get the permutation matrix Π via $G\Pi = QR$, where *Q* is an unitary matrix, *R* is an upper triangular matrix. The absolute values of the diagonal elements of *R*, denoted as $|R_{ii}|$, decrease as *i* increases and are named as *R*-values.
2) Rank fuzzy rules in terms of the *R*-values and the permutation matrix Π in which each column has one element taking value 1 and all the other elements taking value 0. Each column of Π corresponds to a fuzzy rule. The numbering of the rule that corresponds to the *j*th column is the same as the numbering of the row where the "1" element of the *j*th column is located. For example, if the "1" of the 1st column is in the 4th row, then the 4th rule is the most important one and its importance is measured as $|R_{11}|$. The rule corresponding to the first column is the most important, and in descending order the rule corresponding to the last column is the least important.

By applying the pivoted QR decomposition algorithm to the induced fuzzy classifier, each rule can be assigned a *R*-value, which measures the importance of the fuzzy rule. However, the *R*-values reflect the rule base structure only, without considering the output contribution of the induced fuzzy rules. Two new indices based on the L2-SVM learning results are proposed in the following.

B. α -values of Fuzzy Rules

It can be seen from the induction procedure described in the above section that for each induced fuzzy rule, its associated Lagrangian multiplier $\tilde{\alpha}_0^{(i)}$ determines the depth of the effect of the rule consequent. Hence $\tilde{\alpha}_0^{(i)}$ is a very useful index for measuring the output contribution of the induced fuzzy rule. These Lagrangian multipliers are called α -values of fuzzy rules in this paper.

C. ω -values of Fuzzy Rules

Although the fuzzy rule ranking by α -values takes into account the output contribution of induced fuzzy rules, it ignores the rule base structure. In order to consider both the rule base structure and the output contribution of fuzzy rules, a ω -value for $Rule_i$ is suggested as follows:

$$\omega_i = \frac{\tilde{\alpha}_o^{(i)} \cdot |R_{ii}|}{\max_i \tilde{\alpha}_o^{(i)} \cdot \max_i |R_{ii}|} \tag{24}$$

where $\tilde{\alpha}_o^{(i)}$ and $|R_{ii}|$ are the α-value and *R*-value of $Rule_i$ respectively.

D. Fuzzy Rule Selection

Given a fuzzy classifier $FC_{SVM}^{(0)}$ induced by the L2-SVM learning process, the α-values and ω-values can be used to identify the most influential fuzzy rules that ensure the smallest possible model that explains the available data well. Let *V* be the validation dataset and *T* the test dataset. The fuzzy rule selection procedure is described by the following steps:

Step 1. Evaluate the misclassification rates (MRs) of the $FC_{SVM}^{(0)}$ on the validation dataset *V* and the test dataset *T* separately, which are represented as $err_V(0)$ and $err_T(0)$;

Step 2. Set *s*=1 and assign a small value to threshold $h_s (h_s > 0)$;

Step 3. Select the most influential fuzzy rules by $\left\{ Rule_{i^*} \,\middle|\, \tilde{\alpha}_o^{(i^*)} \ or \ \omega_{i^*} > h_s \right\}$;

Step 4. Construct a fuzzy classifier $FC_{SVM}^{(s)}$ by using the influential fuzzy rules selected in *Step 3;*

Step 5. Apply $FC_{SVM}^{(s)}$ to the validation dataset *V* and the test dataset *T* to obtain new MRs: $err_V(s)$ and $err_T(s)$;

Step 5. If $err_V(s) > err_V(0)$, stop the selection and use $FC_{SVM}^{(s-1)}$ as the final compact classifier and $err_T(s-1)$ as the measure of generalization performance for $FC_{SVM}^{(s-1)}$; Otherwise, increase *s* by 1, assign a higher value to threshold h_s, and go to *Step 3.*

E. Implementation of the Proposed L2-SVM Based Fuzzy Classifier Construction

Given a training dataset $\left\{ x^{(l)}, y^{(l)} \right\}_{l=1}^{N}$, where $y^{(l)} \in \{-1, 1\}$, the proposed scheme includes the following steps:

Step 1. Initialization

Assign the same small value to parameters φ_j, i.e., treat each feature equally at the beginning; Assign an initial value to the regularization parameter *C* and a small positive value to ε_J;

Step 2. Perform the L2-SVM learning to obtain the optimal solution α_0 and the margin $1/(2W(\alpha_0, \theta))$;

Step 3. Solve the quadratic programming problem (18)~(19) to get the optimal solution β_0 and the squared radius $S(\theta)$;

Step 4. Update parameters φ_j and *C* in terms of the gradients of *J* with respect to φ_j and *C* separately, and update θ_j according to (8);

Step 5. Go to *Step 2* until the radius-margin bound decrement $\Delta J < \varepsilon_J$;

Step 6. Extract fuzzy rules as indicated in section II-*B*.

*Step 7.*Conduct fuzzy rule ranking and rule subset selection as indicated in subsections III-B, C, and D to obtain a more compact fuzzy classifier.

IV. Experimental Results

In this section we evaluate the performance of the proposed L2-SVM based fuzzy classification algorithm on the benchmark problems based on ringnorm data and german data, which are available from the DELVE repository (*http://www.cs.toronto.edu/~delve/data/ringnorm/*) and the UCI repository (http://www.ics.uci.edu/~mlearn/MLRepository.html), in comparison with some well-known fuzzy and non-fuzzy classifiers.

A. Experiments on Ringnorm Dataset

The ringnorm dataset contains 7400 samples, each consisting of 20 attributes (features). This is a 2-class classification problem proposed by Breiman who reported that the theoretically expected MR is 1.3% [29]. For such a high-dimensional problem, it is very difficult to apply grid partitioning to generate fuzzy rules. Imagine that in the simplest case, if two fuzzy sets were used to partition each attribute, then a grid partitioning based method would generate 2^{20} fuzzy rules. However, prototype based fuzzy classifiers like the proposed one can avoid this dilemma of dimensionality.

In the following experiment, the radius-margin bound is normalized as $J_N = 2S(\theta)W(\alpha_0, \theta)/N$, and the parameter *C* is updated by using a transform $u = \log(C), \partial/\partial C = \partial/\partial u \cdot 1/C$ to meet the requirement of $C > 0$. *C* was initially set to 1, parameters θ_j were initialized to be 0.5 by setting the initial value of φ_j as 0.8326, the learning rates for updating *u* and φ_j were set as 0.0001 and 0.01 separately, and the threshold for updating the radius-margin bound was set as $\varepsilon_J = 5 \times 10^{-5}$. From the available

7400 ringnorm samples, 400 samples were randomly selected for the training process, 5000 samples for the testing process, and the remaining 2000 samples as a validation dataset for fuzzy rule subset selection. After the L2-SVM learning process, 249 support vectors were generated, that is, 249 fuzzy rules were generated for the induced fuzzy classifier. The induced fuzzy classifier produced 66 misclassifications on the test dataset with a MR of 1.32%, which shows that the L2-SVM based fuzzy classifier possesses good generalization ability on the ringnorm problem. The feature ranking results are given in Table 1, in which the 20 attributes are sorted by the values of $\widetilde{\theta}_j$ in a descending order.

{To insert Table 1 here}

For the purpose of comparison, one linear classification method and five nonlinear ones were applied to the ringnorm data with the same training set, test set, and validation set. These methods include linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) [30], RBFN with OLS based forward selection (OLS-RBFN) [27], multilayer perceptron (MLP) [28], fuzzy learning vector quantization (FLVQ) [31], and FLVQ combined with a MLP (FLVQ-MLP). The LDA misclassified 1227 samples with a MR of 24.54% on the test dataset, and the QDA produced 130 misclassifications on test samples with a MR of 2.6%, which implies that the ringnorm data is not linearly separable. Based on the function package provided by Orr [32], the OLS-RBFN was implemented to classify the ringnorm data. In our experiment, the bias term was considered in the decision function, and the generalized cross-validation (GCV) was used as a model selection criterion for OLS-RBFN to balance the bias and variance and optimally select a subset of RBFs. The widths of RBFs were also optimized as indicated in [33]. Two types of RBFs were used in our experiment: Gaussian basis functions (BFs) and Cauchy BFs. The OLS-RBFN with Gaussian BFs achieved a MR of 2.52% by misclassifying 126 test samples, and there were 156 misclassifications produced by the OLS-RBFN with Cauchy BFs leading to a MR of 3.12% on the test dataset. The generalized delta rule [28] was used to train the MLP network with 15 hidden neurons, in which the momentum parameter and the learning rate were set as 0.3 and 0.7 separately. The trained MLP misclassified 650 test samples with a MR of 13%.

It is worthily noted that RBFN with Gaussian BFs can be regarded as a sort of fuzzy classifier, as there exists equivalence between fuzzy systems and RBFNs with Gaussian BFs [34][35][36]: i) The number of RBF units is equal to the number of fuzzy IF-THEN-rules; ii) The output of each fuzzy rule is a constant (the fuzzy system is a zero-order TS fuzzy system); iii) The MFs within each fuzzy rule are chosen as Gaussian functions with the same variance in RBFN: iv) The T-norm operator used in fuzzy system to compute the activation of each rule is multiplication. Hence, we treat the above OLS-RBFN with Gaussian BFs as a neural-fuzzy system with automatic model selection. However, in order to further compare the proposed L2-SVM based fuzzy classifier with the well-known fuzzy classifiers, we also applied the FLVQ to the ringnorm problem. FLVQ requires its user to specify the number of prototypes, the initial exponent ex_0, the final exponent ex_f, and the maximum number of epochs. In [31], a heuristic constraint $7 > ex_0 > ex_f > 1.1$ is recommended. As FLVQ is a clustering algorithm, in our experiment the labels were not used in the clustering process, but used for calculating the clustering error rate MR to evaluate the performance of FLVQ. From our experiment, it is found that the choice of the maximum number of epochs had great influence on the performance of FLVQ, whereas the variations of ex_0 and ex_f in the interval $(1.1, 7)$ did not much impact the performance of FLVQ. Hence, the available validation dataset with 2000 samples was used to find an optimal maximum number of epochs for FLVQ, and then the FLVQ was applied to the dataset with 5400 samples including the above training dataset and the test dataset. This is because FLVQ is an unsupervised clustering algorithm which does not need to divide an available dataset into training one and test one. On the ringnorm problem, the trained FLVQ misclassified 1320 samples with a MR of 24.44%. This result is not surprising, because the clustering is completely unsupervised and does not take the given desired output information (class labels) into account, which could become a possible problem of the approach in classification as noted by Bishop [37]. To make a fairer comparison, FLVQ combined with a MLP was tested, in which the FLVQ worked as a feature extractor in the first stage and the MLP as classifier in the second stage. This FLVQ-MLP classifier achieves a MR of 2.46% by misclassifying 123 test samples.

{To insert Table 2 here}

The above classification results are summarized in Table 2, which shows that the L2-SVM based fuzzy classifier outperforms the well-known pattern classification methods. Another important objective of our experiments on the ringnorm problem is to test the effectiveness of the proposed fuzzy rule selection method. Therefore, after the L2-SVM learning fuzzy rule ranking was conducted in terms of the *R*-values, α-values, and ω-values of fuzzy rules separately. Fig. 1, Fig. 2, and Fig. 3 illustrate the *R*-values, α-values, and ω-values of the induced fuzzy rules respectively. It can be seen that each rule has different *R*-value, α-value, and ω-value relatively, and that a rule with higher *R*-value does not mean it definitely has higher α-value or ω-value, and vice versa. Hence, these three indices evaluate the importance of fuzzy rules in their own ways. In the following, in order to construct the possible smallest classifier with good generalization performance, these three indices were separately used to select the most influential fuzzy rules.

Table 3 summarizes the rule subset selection results, in which $h_s = 0$ corresponds to applying the initially induced fuzzy classifier $FC_{SVM}^{(0)}$ (i.e., without rule selection). It can be seen that in terms of *R*-values, the smallest fuzzy classifier, which keeps the MR of 1.32% on test samples, consists of 214 fuzzy rules. In terms of α -values of fuzzy rules, taking into account the effects of rule consequents, the rule selection procedure identified 90 most influential fuzzy rules with the MR of 1.32% reserved on the test dataset. Similar rule selection result was obtained by using ω-values of fuzzy rules, which produced a fuzzy classifier with 89 rules and led to the MR of 1.32% on test samples. As a matter of fact, fuzzy rule ranking in terms of R-values is based on the QR decomposition method [17]. It is clear that the proposed scheme using α-values and ω-values outperforms the QR decomposition method in fuzzy rule selection by identifying much more compact rule bases.

{To insert Fig. 1 here}

{To insert Fig. 2 here}

{To insert Fig. 3 here}

{To insert Table 3 here}

In order to demonstrate the effect of dropping the least important features on fuzzy rule selection and classification performance, 18 firstly ranked features were selected to construct the fuzzy model. Before fuzzy rule selection, the induced fuzzy classifier based on the 249 fuzzy rules using 18 features, denoted as $\overline{FC}_{SVM}^{(0)}$, achieved a MR of 2.06% on test samples. It is noted that the α-values of fuzzy rules in $\overline{FC}_{SVM}^{(0)}$ are the same as the ones in $FC_{SVM}^{(0)}$, whereas the *R*-values and ω-values of the fuzzy rules in $\overline{FC}_{SVM}^{(0)}$ are different from the ones in $FC_{SVM}^{(0)}$ respectively. Table 4 gives the corresponding rule subset selection results using 18 features in terms of the *R*-values, α-values, and ω-values of the fuzzy rules in $\overline{FC}_{SVM}^{(0)}$. The smallest fuzzy classifier produced by using *R*-values, with the MR of 2.06% on test samples, contains 230 fuzzy rules. By using α-values of fuzzy rules in the rule selection procedure, 89 fuzzy rules were selected for the induced fuzzy classifier with the MR of 2.06% on test samples. As shown in Table 4, in terms of ω-values of fuzzy rules, a fuzzy classifier with 83 fuzzy rules was induced and achieved a MR of 1.78% on test samples, which is better than using all the 249 fuzzy rules.

{To insert Table 4 here}

B. Experiments on German Dataset

In the following, the german dataset was used to further evaluate the proposed method in comparison with the well-known classifiers. The german credit dataset with 1000 samples is known as a benchmark problem for its 2 classes with many odd samples in 20-dimensional space. In this experiment, 300 samples were randomly selected for training L2-SVM, 400 samples for testing process, and the remaining 300 samples as a validation dataset for fuzzy rule subset selection. The learning rates for updating u and φ_j were set as 0.00005 and 0.006 separately by a trial and error approach, parameters θ_j were all initialized as 0.5 with initial φ_j=0.8326, C was initialized to 1, and the threshold for updating radius-margin bound was set as $\varepsilon_{J_N} = 6 \times 10^{-4}$. After the L2-SVM learning process, 195 support vectors were generated, that is to say, 195 fuzzy rules were generated for the induced fuzzy classifier, which produced 98 misclassifications on the test dataset with a MR of 24.5%. Similar to the experiment with the ringnorm dataset, the well-known classification methods, LDA, QDA, OLS-RBFN, MLP, FLVQ, and FLVQ-MLP, were used to compare with the L2-SVM induced fuzzy classifier. The LDA misclassified 125 samples with a MR of 31.25% on the test dataset, and the QDA produced 118 misclassified test samples with a MR of 29.5%. In the experiment on OLS-RBFN, the bias term was also considered in the decision function, and GCV was used as a model selection criterion for OLS-RBFN to balance the bias and variance, and optimally select a subset of RBFs. The widths of RBFs were also optimized. The OLS-RBFN with Gaussian BFs achieved a MR of 28.5% by misclassifying 114 test samples, and there were 110 misclassifications produced by the OLS-RBFN with Cauchy BFs leading to a MR of 27.5% on the test dataset. The MLP network with 15 hidden neurons was trained by the generalized delta rule with momentum parameter 0.3 and the learning rate 0.7. The trained MLP misclassified 112 test samples with a MR of 28%. When FLVQ was applied to a dataset with 700 samples including the above training samples and test samples, it achieves a MR of 37.85%. By combining FLVQ and a MLP with 15 hidden neurons, the momentum parameter as 0.1 and the learning rate 0.2, the FLVQ-MLP classifier misclassified 113 test samples with a MR of 28.25%. The above classification results are summarized in Table 5. As analyzed in [38], there exists too much noise in the german credit dataset, which weakens the predictive capability of the features. From Table 5, it can be seen that the L2-SVM induced fuzzy classifier also outperforms the well-known classification methods in terms of the generalization performances on the german credit problem.

{To insert Table 5 here}

Furthermore, in order to construct the possible smallest classifier with good generalization performance, the three indices: *R*-values, α -values, and ω -values of fuzzy rules were separately used to select the most influential fuzzy rules. Fig.4, Fig.5, and Fig.6 illustrate the *R*-values, α -values, and ω -values of the fuzzy rules respectively. Table 6 summarizes the rule subset selection results. It can be seen that in terms of *R*-values, the smallest fuzzy classifier, which achieves a MR of 25.25% on test samples, consists of 112 fuzzy rules. In terms of α -values, 77 most influential fuzzy rules were identified with a MR of 25.00% on the test dataset. The rule selection by using ω -values produced a fuzzy classifier with 70 rules and led to a MR of 24.75% on test samples.

{To insert Fig. 4 here}

{To insert Fig. 5 here}

{To insert Fig. 6 here}

{To insert Table 6 here}

From the above results it can be seen that the two new indices, α -values and ω -values of fuzzy rules, can generate much more compact rule bases than the classifier generated by the traditional *R*-values and the initially induced classifier $FC_{SVM}^{(0)}$. This indicates that the induced fuzzy rules, corresponding to the support vectors in the SVM, should not be treated equally in the classification even though the inherent mechanism of the SVM has the potential of producing sparse solutions. Some support vectors or fuzzy rules are much more important than the others.

V. DISCUSSIONS

The proposed L2-SVM based fuzzy classifier and the rule ranking indices possess some additional merits that are worthy of being delineated further. One additional merit is that the proposed fuzzy rule ranking indices are also very useful for identifying the most influential support vectors for SVM itself. Although the SVM learning process produces sparse support vectors, it treats the support vectors equally in the classification process in the sense that all support vectors are equally considered for the classification. To the best of our knowledge, currently there is no special mechanism to select the most influential support vectors by considering the different depths of real contributions to the classification from different support vectors. A potential problem is that there may be redundant or correlated support vectors in the SVM. If a support vector ranking is produced for the SVM classification according to the importance of support vectors, a more parsimonious SVM classifier can be obtained in terms of the ranking results. Because each fuzzy rule in the induced fuzzy classifier corresponds to a support vector in L2-SVM, the proposed two rule ranking indices can be directly used to identify the most influential support vectors for SVM classification.

The second additional merit is that the proposed method provides a new way of constructing prototype based fuzzy classifiers, which is different from the most currently used prototype based fuzzy classifiers. It is known that the outstanding advantage of prototype based fuzzy classifiers over grid based fuzzy classifiers lies in that the prototype based fuzzy classifiers can overcome the curse of dimensionality. However, there are three fundamental issues needed to be addressed in designing a prototype-based classifier [39]: i) How many prototypes are to be generated; ii) How to generate the prototypes; and iii) How to use the prototypes to design a classifier. Currently, in most efforts made to design prototype based fuzzy classifiers, these three issues are addressed independently and separately. For example, unsupervised clustering algorithms such as c-means [40], fuzzy c-means [41], and FLVQ [31], are widely used to generate prototypes, but most of the clustering algorithms require the number of clusters (prototypes) to be supplied externally or to be determined by using some cluster validity indices. Once the prototypes are generated, there are different ways of using the prototypes to design the classifier. One commonly used strategy is that these currently generated prototypes are used as initial fuzzy partitions, and an adaptive learning algorithm such as neural network learning algorithm is then applied to update these prototypes. Finally, based on training dataset, the adaptive prototype based classifier is trained optimally with good generalization performance on test dataset. An example of using this strategy is the neuro-fuzzy classifier NEFCLASS [42], which uses fuzzy clustering to initialize its prototypes in [43]. Recently, new efforts have been made to develop prototype based classifiers by integrating the above issues into one modeling process. Mountain clustering method can automatically estimate the number of prototypes whilst generating the prototypes [44]. Laha and Pal [39] suggested two approaches to designing nearest prototype classifiers by addressing the problem of finding the required number of prototypes as well as the prototypes themselves together. The proposed method of constructing fuzzy classifiers based on L2-SVM in this paper can fulfill the integration of all the three issues together. In the L2-SVM based fuzzy classifier, one does not need to specify the initial number of fuzzy rules in advance, because each fuzzy rule corresponds to a support vector, and the number of support vectors or fuzzy rules depends on the number of non-zero Lagrangian multipliers $\alpha^{(i)}$. After the L2-SVM learning process, not only the support vectors, i.e., the prototypes, are generated, its classifier defined by the support vectors in a decision surface is also produced. These Lagrangian multipliers are naturally obtained from solving a quadratic programming. That is to say, all the above three issues in designing a prototype based classifier are addressed together

and automatically identified from data in one model structure in the proposed scheme.

The third additional advantage of the proposed method is that not only the fuzzy rules are generated optimally from data through the SVM learning, but also the ranking results of all the input features are simultaneously obtained. Although traditional methods for feature ranking are capable of identifying the influential features for fuzzy modeling [45][46], most of them perform feature ranking in a separate phase from the classifier construction process. Importantly, if feature ranking and classifier construction are performed simultaneously in an integrated way, the goodness of features can be learned automatically from data and the most appropriate set of features relevant to the task could be found [47][48]. As a result, a parsimonious fuzzy model with good generalization performance would be obtained. In the proposed L2-SVM based fuzzy classification system, after the training process, an importance rank of each feature is discovered and the values of parameters characterizing MFs can also be evaluated based on the feature ranking results. In such a way, both feature ranking and automatic updating of MF parameters can be realized in an integrated manner.

VI. Conclusion

In this paper a new scheme is proposed for constructing parsimonious fuzzy classifiers with simultaneous model selection and feature ranking based on the L2-SVM technique. Another contribution of this paper is to have proposed two novel indices, α-values and ω-values of fuzzy rules, for fuzzy rule selection based on the L2-SVM learning results. Because the number of induced fuzzy rules in the L2-SVM based fuzzy classifier is not related to the dimensionality of input space, the proposed scheme provides an efficient way of avoiding the "curse of dimensionality" during constructing fuzzy classifiers in high-dimensional space. Furthermore, the combination of model selection, feature ranking, and fuzzy rule selection in the proposed scheme leads to parsimonious fuzzy classifier construction, which is demonstrated by experiments on two benchmark high-dimensional problems. The experimental results have also shown that α-values and ω-values are more effective than the traditional *R*-values in fuzzy rule ranking and selection.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions which have helped improve this paper.

References

[1] V. Vapnik, *Statistics Learning Theory*, John Wiley & Sons, 1998.
[2] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. on Neural Networks*, vol. 13, no. 2, pp. 464-471, 2002.
[3] J-H. Chiang and P.-Y. Hao, "A new kernel-based fuzzy clustering approach: support vector clustering with cell growing," *IEEE Trans. on Fuzzy Systems*, vol. 11, no. 4, pp. 518-527, 2003.
[4] S. M. Zhou and J. Q. Gan, "An unsupervised kernel based fuzzy c-means clustering algorithm with kernel normalization," *Int. Journal of Computational Intelligence and Applications*, vol. 4, no. 4, pp. 355-373, 2004.
[5] Y. Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems*, vol. 11, no. 6, pp. 716-728, 2003.
[6] D. Dubois and H. Prade, "Operations on fuzzy numbers," *Int. Journal of Syst. Sci.*, vol. 9, no. 6, pp. 613–626, 1978.
[7] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, no. 4, pp. 637–649, 1998.
[8] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
[9] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Trans. on Neural Networks*, vol. 13, no. 3, pp. 780-784, 2002.
[10] S. W. Kim and B. J. Oommen, "On utilizing search methods to select subspace dimensions for kernel-based nonlinear subspace classifiers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 136-141, 2005.
[11] F. Camastra and A. Verri, "A novel kernel method for clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 801-805, 2005
[12] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131-159, 2002.
[13] G. C. Mouzouris and J. M. Mendel, "Designing fuzzy logic systems for uncertain environments using a singular-value-QR decomposition method," *Proc. 5th IEEE Int. Conf. on Fuzzy Syst.*, New Orleans, LA, Sept. 1996, pp. 295–301.
[14] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Trans. on Fuzzy Systems*, vol. 6, no. 4, pp. 530-537, 1998.
[15] J. Yen and L. Wang, "Application of statistical information criteria for optimal fuzzy model construction," *IEEE Trans. on Fuzzy Syst.*, vol. 6, no. 3, pp. 362–372, 1998.
[16] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. on Syst., Man, Cybern. -Part B*, vol. 29, no. 1, pp. 13–24, 1999.
[17] M. Setnes and R. Babuska, "Rule base reduction: some comments on the use of orthogonal transforms," *IEEE Trans. on Sys., Man, Cybern.-Part C*, vol. 31, no. 2, pp. 199-206, 2001.
[18] G. H. Golub and C. F. van Loan, *Matrix Computations* (2nd ed.), Baltimore, MD: Johns Hopkins Univ. Press, 1989.
[19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control," *IEEE Trans. on Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116-132, 1985.
[20] M. Setnes, R. Babuska, and H. B. Verbruggen, "Rule-based modelling: precision and transparency," *IEEE Trans. on Syst., Man, Cybern. -Part C*, vol. 28, no. 1, pp. 165-169, 1998.
[21] C. Distante, N. Ancona, and P. Siciliano, "Support Vector Machines for olfactory signals recognition," *Sensors and Actuators B*, vol. 88, no. 1, pp. 30-39, 2003.
[22] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.
[23] S. McLoone, M. D. Brown, G. Irwin, and G. Lightbody, "A hybrid linear/nonlinear training algorithm for feedforward neural networks," *IEEE Trans. Neural Networks*, vol. 9, no. 4, pp. 669–684, 1998.
[24] B. A. Whitehead and T. D. Choate, "Evolving space-filling curves to distribute radial basis functions over an input space," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 15–23, 1994.
[25] Z. R. Yang and S. Chen, "Robust maximum likelihood training of heteroscedastic probabilistic neural networks," *Neural Networks*, vol. 11, no. 4, pp. 739–747, 1998.
[26] H. Peng, T. Ozaki, V. Haggan-Ozaki, and Y. Toyoda, "A parameter optimization method for radial basis function type models," *IEEE Trans. on Neural Networks*, vol. 14, no. 2, pp. 432-438, 2003.

[27] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning for radial basis function networks", *IEEE Trans. on Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.
[28] S. Haykin, *Neural Networks: A Comprehensive Foundation* (2nd ed), Prentice-Hall, Inc, 1999.
[29] L. Breiman, "Arcing classifiers," *Annals of Statistics*, vol. 26, no. 3, pp. 801-849, 1998.
[30] G. J. McLachlan, *Discriminant Analysis and A statistical Pattern Recognition*, NY: John Wiley, 1992.
[31] J. C. Bezdek and N. R. Pal, "Two soft relative of learning vector quantization," *Neural Networks*, vol. 8, no. 5, pp. 729-743, 1995.
[32] M. Orr, www.anc.ed.ac.uk/~mjo/rbf.html
[33] M. Orr, "Optimising the widths of RBFs," *Proc. of the Fifth Brazilian Symposium on Neural Networks*, Belo Horizonte, Brazil, 1998.
[34] B. Fritzke, "Incremental neuro-fuzzy systems," *Proc. of Applications of Soft Computing, SPIE International Symposium on Optical Science, Engineering and Instrumentation*, San Diego, USA, 1997.
[35] Y. Jin and B. Sendhoff, "Extracting interpretable fuzzy rules from RBF networks," *Neural Processing Letters*, vol. 17, no. 2, pp. 149-164, 2003.
[36] K. J. Hunt, R. Haas, and R. Murray Smith, "Extending the functional equivalence of radial basis function networks and fuzzy inference systems," *IEEE Trans. on Neural Networks*, vol. 7, no. 3, pp. 776-781, 1996.
[37] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
[38] P. O'Dea, J. Griffith, and C. O'Riordan, "Combining feature selection and neural networks for solving classification problems," *Proc. of the 12th Irish conference on Artificial Intelligence and Cognitive Science,* September, 2001.
[39] A. Laha and N. R. Pal, "Some novel classifiers designed using prototypes extracted by a new scheme based on self-organizing feature map," *IEEE Trans. on Sys., Man, Cybern.-Part B*, vol. 31, no. 6, pp. 881-890, 2001.
[40] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1974.
[41] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[42] D. Nauck and R. Kruse, "NEFCLASS-A neuro-fuzzy approach for the classification of data," *Proc. of the 1995 ACM Symposium on Applied Computing*, Nashville, USA, February 1995, pp. 461-465.
[43] D. Nauck and F. Klawonn, "Neuro-fuzzy classification initialized by fuzzy clustering," *Proc. of the 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT)*, Aachen, Germany, 1996.
[44] R. Yager and D. Filev, "Generation of fuzzy rules by mountain clustering," *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3, pp. 209-219, 1994.
[45] R. De, N. R. Pal, and S. K. Pal, "Feature analysis: neural network and fuzzy set theoretic approaches," *Pattern Recognition*, vol. 30, no. 10, pp. 1579–1590, 1997.
[46] D. Tikk, T. D. Gedeon, and K. W. Wong, "A feature ranking algorithm for fuzzy modeling problems," In J. Casillas, O. Cordón, F. Herrera, L. Magdalena (eds.), *Interpretability Issues in Fuzzy Modeling,* number 128 in *Studies in Fuzziness and Soft Computing*, Springer–Verlag, Heidelberg, 2003, pp.176–192.
[47] D. Chakraborty and N. R. Pal, "A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification," *IEEE Trans. on Neural Networks*, vol. 15, no. 1, pp. 110-123, 2004.
[48] D. Chakraborty and N. R. Pal, "Integrated feature analysis and fuzzy rule-based system identification in a neuro-fuzzy paradigm," *IEEE Trans. on Syst., Man, Cybern. -Part B*, vol. 31, no. 3, pp. 391-400, 2001.

Shang-Ming Zhou (M'01) received the BSc. degree in mathematics from Liaocheng University, Shandong Province, China, and the MSc. degree in applied mathematics from Beijing Normal University, Beijing, China, in 1985 and 1992 respectively.

He worked as an Associate Professor at China Remote Sensing Satellite Ground Station, Chinese Academy of Sciences, from 1998 to 2002. Currently he is with Department of Computer Science at University of Essex, UK. His research interests include machine learning, complex system modelling, interpretability of neuro-fuzzy systems, computational intelligence, intelligent signal/image processing and data mining.

John Q. Gan (SM'01) received the B.Sc. degree in electronic engineering from Northwestern Polytechnic University, China, in 1982, the M.Eng. degree in automatic control and the Ph.D. degree in biomedical electronics from Southeast University, China, in 1985 and 1991, respectively.

He is a Senior Lecturer in the Department of Computer Science, the University of Essex, U.K. He has co-authored a book and published over 100 research papers. His research interests are in neurofuzzy computation, brain-computer interfaces, robotics and intelligent systems, pattern recognition, signal processing, and data fusion.

Captions of Figures

Fig. 1 *R*-values of induced fuzzy rules using 20 features of ringnorm data

Fig. 2 α-values of induced fuzzy rules using 20 features of ringnorm data

Fig. 3. ω-values of induced fuzzy rules using 20 features of ringnorm data

Fig. 4 *R*-values of induced fuzzy rules using 20 features of german data

Fig. 5 α-values of induced fuzzy rules using 20 features of german data

Fig. 6 ω-values of induced fuzzy rules using 20 features of german data

Captions of Tables

Table 1 Feature Ranking in the Descending Order for the 20 Features of Ringnorm Data

Table 2 Generalization Performances of the Well-Known Algorithms on Ringnorm Data

Table 3 Fuzzy Rule Subset Selection in Terms of *R*-Values, α-Values, and ω-Values of Fuzzy Rules Using 20 Features of Ringnorm Data

Table 4 Fuzzy Rule Subset Selection in Terms of *R*-Values, α-Values, and ω-Values of Fuzzy Rules Using 18 Features of Ringnorm Data

Table 5 Generalization Performances of the Well-Known Algorithms on German Data

Table 6 Fuzzy Rule Subset Selection in Terms of *R*-Values, α-Values, and ω-Values of Fuzzy Rules Using 20 Features of German Data

Figures

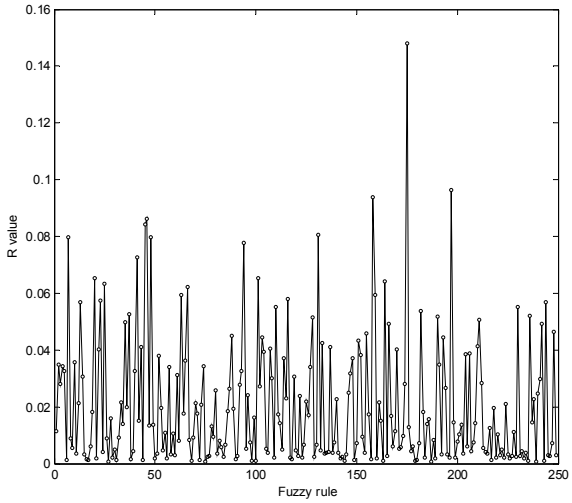

Fig. 1 *R*-values of induced fuzzy rules using 20 features of ringnorm data

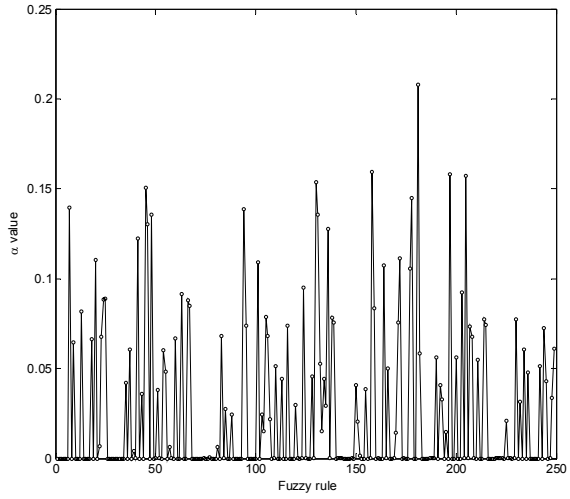

Fig. 2 α -values of induced fuzzy rules using 20 features of ringnorm data

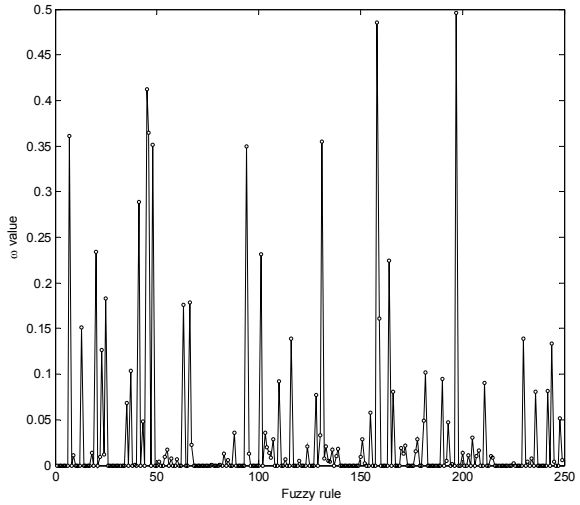

Fig. 3. ω -values of induced fuzzy rules using 20 features of ringnorm data

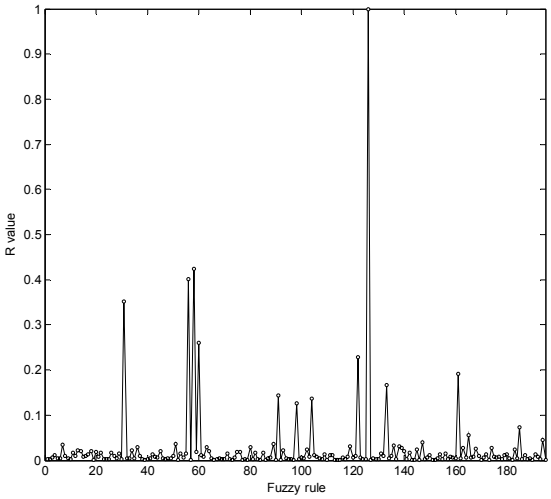

Fig. 4 *R*-values of induced fuzzy rules using 20 features of german data

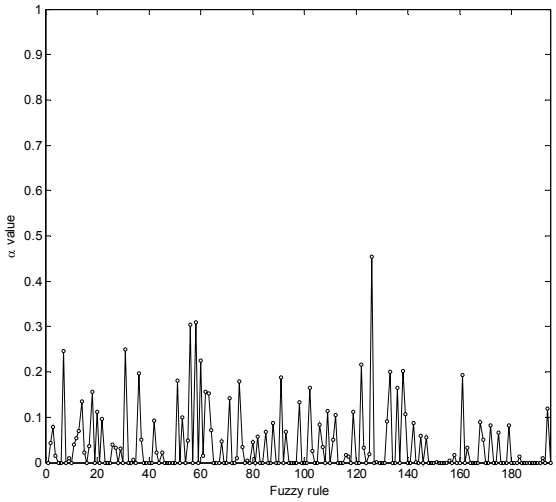

Fig. 5 α -values of induced fuzzy rules using 20 features of german data

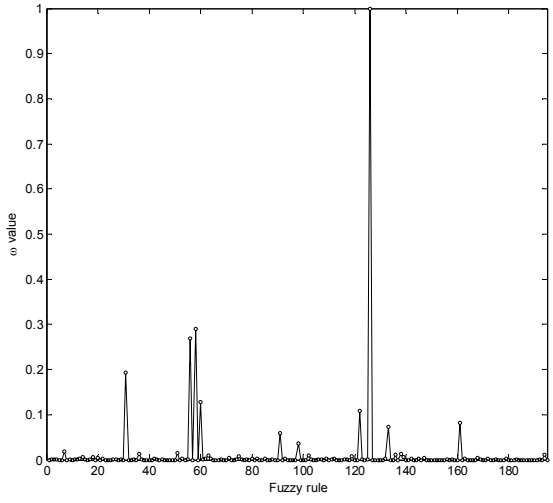

Fig. 6 ω -values of induced fuzzy rules using 20 features of german data

Tables

TABLE 1
FEATURE RANKING IN THE DESCENDING ORDER FOR THE 20 FEATURES OF RINGNORM DATA

Features	9	1	8	15	12	5	10	2	3	20
Feature ranking by $\tilde{\theta}$	1.000	0.941	0.858	0.840	0.826	0.825	0.81	0.78	0.77	0.76
Features	6	18	13	16	14	7	4	11	19	17
Feature ranking by $\tilde{\theta}$	0.755	0.754	0.752	0.751	0.739	0.737	0.72	0.653	0.65	0.64

TABLE 2
GENERALIZATION PERFORMANCES OF THE WELL-KNOWN ALGORITHMS ON RINGNORM DATA

Methods	LDA	QDA	OLS-RBFN with Gausian BFs	OLS-RBFN with Cauchy BFs	MLP	FLVQ	FLVQ-MLP	The proposed
MRs	24.54%	2.6%	2.52%	3.12%	13%	24.44%	2.46%	1.32%

TABLE 3
FUZZY RULE SUBSET SELECTION IN TERMS OF *R*-VALUES, α-VALUES, AND ω-VALUES OF FUZZY RULES USING 20 FEATURES OF RINGNORM DATA

Using *R*-value index				Using α-value index				Using ω-value index			
h_s	No. of rules selected	err_V	err_T	h_s	No. of rules selected	err_V	err_T	h_s	No. of rules selected	err_V	err_T
0	249	1.45%	1.32%	0	249	1.45%	1.32%	0	249	1.45%	1.32%
0.001	242	1.45%	1.32%	**0.001**	**90**	**1.45%**	**1.32%**	0.0001	90	1.45%	1.32%
0.002	**214**	**1.45%**	**1.32%**	0.002	89	1.50%	1.32%	**0.0006**	**89**	**1.45%**	**1.32%**
0.003	193	1.80%	1.5%	0.005	88	1.55%	1.38%	0.0008	88	1.50%	1.34%

TABLE 4
FUZZY RULE SUBSET SELECTION IN TERMS OF *R*-VALUES, α-VALUES, AND ω-VALUES OF FUZZY RULES USING 18 FEATURES OF RINGNORM DATA

Using *R*-value index				Using α-value index				Using ω-value index			
h_s	No. of rules selected	err_V	err_T	h_s	No. of rules selected	err_V	err_T	h_s	No. of rules selected	err_V	err_T
0	249	2.05%	2.06%	0	249	2.05%	2.06%	0	249	2.05%	2.06%
0.001	**230**	**2.05%**	**2.06%**	0.001	90	2.05%	2.06%	0.0001	90	2.05%	2.06%
0.002	197	2.15%	1.92%	**0.002**	**89**	**2.05%**	**2.06%**	0.0006	88	2.00%	2.02%
				0.005	88	2.10%	2.04%	0.0016	87	1.90%	2.02%
								0.002	84	1.85%	1.94%
								0.0025	**83**	**1.85%**	**1.78%**
								0.0035	82	2.30%	1.84%

TABLE 5
GENERALIZATION PERFORMANCES OF THE WELL-KNOWN ALGORITHMS ON GERMAN DATA

Methods	LDA	QDA	OLS-RBFN with Gausian BFs	OLS-RBFN with Cauchy BFs	MLP	FLVQ	FLVQ-MLP	The proposed
MRs	31.25%	29.5%	28.5%	27.5%	28%	37.85%	28.25%	24.5%

TABLE 6
FUZZY RULE SUBSET SELECTION IN TERMS OF *R*-VALUES, α-VALUES, AND ω-VALUES OF FUZZY RULES USING 20 FEATURES OF GERMAN DATA

Using *R*-value index				Using α-value index				Using ω-value index			
h_s	No. of rules selected	err_V	err_T	h_s	No. of rules selected	err_V	err_T	h_s	No. of rules selected	err_V	err_T
0	195	27.33%	24.50%	0	195	27.33%	24.50%	0	195	27.33%	24.50%
0.001	182	27.33%	24.50%	0.001	81	27.33%	24.50%	0.00001	81	27.33%	24.50%
0. 004	126	27.33%	26.00%	**0.01**	**77**	**27.33%**	**25.00%**	0.00015	75	26.67%	25.00%
0.0045	118	27.33%	25.75%	0.0102	76	28.00%	25.50%	0.0003	71	27.00%	25.25%
0.0049	113	27.33%	25.25%					**0.00034**	**70**	**27.00%**	**24.75%**
0.00498	**112**	**27.33%**	**25.25%**					0.00035	69	28.33%	25.50%
0.005	111	28.67%	26.25%								