Pattern Recognition 43 (2010) 3712–3729

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

SVM-FuzCoC: A novel SVM-based feature selection method using a fuzzy complementary criterion

S.P. Moustakidis, J.B. Theocharis *

Aristotle University of Thessaloniki, Department of Electrical & Computer Engineering, Auth University Campus, 54124 Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 27 January 2009
Received in revised form
5 February 2010
Accepted 4 May 2010

Keywords:
Feature selection
Fuzzy sets
Feature redundancy
Fuzzy complementary criterion
Support vector machines

ABSTRACT

An efficient filter feature selection (FS) method is proposed in this paper, the SVM-FuzCoC approach, achieving a satisfactory trade-off between classification accuracy and dimensionality reduction. Additionally, the method has reasonably low computational requirements, even in high-dimensional feature spaces. To assess the quality of features, we introduce a local fuzzy evaluation measure with respect to patterns that embraces fuzzy membership degrees of every pattern in their classes. Accordingly, the above measure reveals the adequacy of data coverage provided by each feature. The required membership grades are determined via a novel fuzzy output kernel-based support vector machine, applied on single features. Based on a fuzzy complementary criterion (FuzCoC), the FS procedure iteratively selects features with maximum additional contribution in regard to the information content provided by previously selected features. This search strategy leads to small subsets of powerful and complementary features, alleviating the feature redundancy problem. We also devise different SVM-FuzCoC variants by employing seven other methods to derive fuzzy degrees from SVM outputs, based on probabilistic or fuzzy criteria. Our method is compared with a set of existing FS methods, in terms of performance capability, dimensionality reduction, and computational speed, via a comprehensive experimental setup, including synthetic and real-world datasets.

1. Introduction

The exponential growth of computer technology in recent years has led to the proliferation of vast amount of data. Since continued data accumulation is inevitable, preprocessing techniques are essential to keep pace with data collection rate. In this context, feature selection (FS) algorithms have become indispensable components of data preprocessing in the field of machine learning [1], statistical pattern recognition [2,3], and data mining [4,5]. The primary goal of FS is to select a subset of valuable input variables by discarding irrelevant or redundant features. Beneficial effects of FS techniques are learning acceleration and improvement of the classifiers' performance. Additionally, dimensionality reduction of feature spaces enhances interpretability of obtained models, since classifiers involving many features are less comprehensive.

A typical FS process initiates by producing candidate feature subsets based on a search strategy. Each candidate subset is evaluated and compared with the previous best one according to a particular evaluation criterion. Depending on the evaluation criterion used, FS methods are divided into three main categories: wrapper, filter, and hybrid models. In wrapper methods, FS adheres to an induction algorithm and the candidate subsets are validated in terms of accuracy provided by a classifier [6]. Many wrapper methods follow a backward process where at each stage, one of the features is excluded from the feature set, the removal of which yields the least reduction in training accuracy [7]. Although high classification rates are usually achieved, wrappers are computationally intense due to their coupling with the classifier. An additional shortcoming is that when dealing with small size datasets, these methods suffer from overfitting and inferior generalization results.

Filter methods are independent of the classification model used. FS in these methods relies on intrinsic characteristics of features to reveal their discriminating power. Along this direction, several measures of relevance have been employed to carry out FS. Correlation criteria are the simplest measures used [8,9], which can detect only linear dependencies of features. Recently, some FS methods are suggested [10–12], utilizing the mutual information metric, to assess the feature relevance to target classes and redundancy between features, although they require greater computational efforts for their implementation. Moreover, *FFSEM* [13] and filter methods presented in [14,15] use class similarity measures with respect to the selected subset as evaluation criteria. *ReliefF* [16] validates the importance of features according to the separability of neighboring patterns. Zhang et al. [17] proposed a feature selection method according to features' constraint preserving ability. More concrete, a 'good'

* Corresponding author.
E-mail address: theochar@eng.auth.gr (J.B. Theocharis).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.05.007

feature should be the one on which two samples from the same class (must-link constraint) are close to each other, whereas samples from different classes (cannot-link constraint) are far away from each other. A recently proposed unsupervised FS method, referred to here as *Mitra* ([18]), partitions the initial feature set into a number of homogeneous subsets and proceeds to selecting a representative feature from each subset. Despite their low complexity, filter approaches do not always succeed with high classification rates as the evaluation criterion used for FS is not necessarily associated with the classifiers to be applied. Hybrid methods [19,20] integrate FS within the learning algorithm, with the goal to exploit the advantages of both wrapper and filter approaches.

Different strategies have been recently investigated for subset generation. *Branch and Bound* (BB) [21] performs an almost exhaustive search but is exponentially prohibitive even with a moderate number of features. *Sequential forward selection* (SFS) [22], which is simpler and faster for moderate dimensionality problems, iteratively adds features to an initial subset so that a given evaluation criterion is maximized, whereas *sequential backward selection* (SBS) eliminates one feature at a time that exhibits the smallest criterion decrease. *Sequential floating forward selection* (SFFS) [23] and *Plus l–take away r* [24] are more sophisticated versions. At each stage, these methods enlarge feature set using forward selection and then, discard features using backward selection. The relevant metrics used in the existing FS methods are geared towards identifying informative features individually and minimizing redundancy present in the reduced feature subset. Nevertheless, they manipulate information globally, in that a single scalar metric is usually employed (relevance index, correlation, redundancy), integrating all patterns of every class.

In this paper, we suggest the SVM-FuzCoC approach, suitable for high-dimensional feature sets. The proposed FS is a filter method and its main characteristics are described as follows. (i) We introduce the notion of fuzzy partition vector (FPV) associated with each feature, which comprises fuzzy membership grades of training patterns (projected on that feature) to their own classes. FPV treats each feature on a pattern-wise base, allowing us to assess redundancy between features. (ii) Exploiting high generalization capabilities of kernel-based support vector machines (K-SVM), a fuzzy output K-SVM (FO-K-SVM) scheme is developed and applied on each single feature, to construct the associated FPV. (iii) The proposed method performs a forward selection guided by a fuzzy complementary criterion (FuzCoC). Particularly, FuzCoC operates on the pre-computed feature FPVs, paying due attention on complementary characteristics between the features. As a result, we obtain small co-operative subsets of discriminating (highly relevant) and non-redundant features, each one covering better a different pattern region. It is analytically shown that FuzCoC acts like a minimal-redundancy–maximal-relevance (mRMR) criterion used by some existing methods of the literature. (iv) FuzCoC-based feature selection does not adhere only to the FO-K-SVM approach. Therefore, several variants of SVM-FuzCoC are developed, whereby apart from the proposed FO-K-SVM, seven other membership degree determination methods are considered. These methods include parametric and non-parametric probabilistic approaches to convert SVM outputs to well calibrated posterior probabilities and methods based on fuzzy criteria. Experimental investigation on a set of benchmark problems of varying complexities shows that computational load of SVM-FuzCoC is reasonably low, while achieving at the same time, a good trade-off between dimensionality reduction and classification accuracy.

The remainder of the paper is organized as follows. Section 2 presents the fuzzy K-SVM classifier used for determining membership grades of patterns over classes. In Section 3, we elaborate on the proposed FS method, including the FPV properties, useful definitions in the fuzzy domain, and presentation of the SVM-FuzCoC algorithm. Section 4 includes illustrative results to highlight attributes of the suggested approach. Comparative results are given in Section 5, contrasting our method with other FS techniques. Additionally, we analyze the performance of SVM-FuzCoC using different schemes for derivation of pattern classification grades. Finally, conclusions are drawn in Section 6.

2. Fuzzy output kernel-based SVM

A fuzzy output kernel-based SVM (FO-K-SVM) approach is suggested in this section, providing both the decision class and the membership grades of patterns to their classes. Initially, we give an outline of K-SVM principles and proceed to a brief review of traditional multiclass extensions. To determine the fuzzy degrees apportioned to classes, binary K-SVM outputs are fuzzified via a properly designed membership function, applied on boundary surfaces.

2.1. Multiclass kernel-based SVM

Consider a dataset comprising labeled training patterns: $D=\{(\boldsymbol{x}_i,y_i), i=1,\ldots,N\}$. Each pattern $\boldsymbol{x}_i=[x_{i,\ 1},x_{i,\ 2},\ldots,x_{i,\ n}]^T\in\Re^n$ belongs to one of two classes, with its class label given as $y_i\in\{+1,-1\}$. Given a nonlinear mapping $\boldsymbol{\Phi}:\Re^n\to\mathcal{F}$, each vector $\boldsymbol{x}_i$ in the original feature space is transformed to a potentially higher dimensional feature space $\mathcal{F}$: $\boldsymbol{x}_i\to\boldsymbol{\Phi}(\boldsymbol{x}_i)$, $i=1,\ldots,N$. The embedding of feature mapping and the associated kernel function lead to a powerful nonlinear scalar-product-based algorithm (K-SVM), executed in$\mathcal{F}$. K-SVM seeks for a suitable separating hyperplane in the transformed space $\mathcal{F}$: $f(x)=((\boldsymbol{w}\cdot\boldsymbol{\Phi}(x))+b)=0$, parameterized by the pair $(\boldsymbol{w},b)$, $\boldsymbol{w}\in\mathcal{F}$, $b\in\Re$ [25,26]. The optimal decision function is obtained by

$$f(\boldsymbol{x})=\mathrm{sgn}\left(\sum_{i\in S}a_iy_iK(\boldsymbol{x}_i,\boldsymbol{x})+b\right) \tag{1}$$

where $S=\{i:0<a_i^*\le C\}$. Coefficient a_i is non-zero when $\boldsymbol{x}_i$ is a support vector; otherwise it is zero. Any function satisfying Mercer's theorem can be used as scalar product, thus serving as a kernel function. In this research, we employ the Gaussian RBF kernel:

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(\frac{-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right) \tag{2}$$

where σ denotes the variance along the feature axis. Selection of the RBF kernel is dictated by its ability to handle high-dimensional data. In our simulations, the parameters C and σ are heuristically determined from a grid of pre-selected values $C\in\{10,30,50,100\}$ and $\sigma\in\{0.005,0.01,0.1,0.5,1\}$.

Since K-SVMs were originally designed for binary classification, a decomposition scheme should be devised to tackle multiclass problems [27]. One-versus-all (OVA) is a common approach, accomplished by combining several binary SVM classifiers. For a M-class problem, OVA proceeds to construct a set of binary classifiers $\{f_1,\ldots f_k,\ldots f_M\}$. Each f_k, $k=1,\ldots,M$, is trained individually to separate class c_k from the rest of the classes, included in $\overline{C}_k=\{1,\ldots,\ell,\ldots,M|\ell\neq k\}$. Following the *winner-takes-all* principle, an unknown pattern $\boldsymbol{x}$ is then assigned to the class that exhibits the maximum decision function value $f_k(\boldsymbol{x})$:

$$c_k=\arg\max_k\left\{f_k(\boldsymbol{x})=\sum_{i\in S}a_i^*y_iK(\boldsymbol{x}_i,\boldsymbol{x})+b^*\right\} \tag{3}$$

2.2. Fuzzy degree determination

Adhering to the OVA decomposition, we introduce here a fuzzy output K-SVM (FO-K-SVM) scheme for multiple class problems. We apply a suitably devised membership function on the output functions $f_k(\boldsymbol{x})$, to exploit local information of patterns in feature space. Initially, a set of M binary K-SVM classifiers is constructed, f_k, $k=1,\ldots,M$, each one separating class k from the rest of the classes. Each classifier f_k assigns a fuzzy grade $\mu_k(\boldsymbol{x}_i)$, indicating the degree to which pattern $\boldsymbol{x}_i$ belongs to class k. The above grade is not derived exclusively by decision value $f_k(\boldsymbol{x})$ (regarded as the primary voter), but in relation to decision values of the competing classifiers $f_\ell(\boldsymbol{x}_i)$, $\ell \in \overline{C}_k$.

Let $m_{i,k}$ denote the maximum decision value among the competing classifiers:

$$m_{i,k} = \max_{l \neq k} f_\ell(\boldsymbol{x}_i) \tag{4}$$

The membership degrees $\mu_k(\boldsymbol{x}_i)$ are determined by applying to every $f_k(\boldsymbol{x}_i)$ the following sigmoid-type function:

$$\mu_k(\boldsymbol{x}_i) = \begin{cases} 0.5 & \text{if } f_k(\boldsymbol{x}_i) = m_{i,k} = 1 \\ \dfrac{1}{1+e^{\left(\ln\left(\frac{1-\gamma}{\gamma}\right)\right)\left[\frac{f_k(\boldsymbol{x}_i)-m_{i,k}}{|1-m_{i,k}|}\right]}} & \text{if } m_{i,k} \neq 1 \end{cases} \tag{5}$$

The rationale underlying the formation of the above membership function rests on the following arguments:

(1) Shape of the membership function is adapted to location of each pattern in regard to decision boundaries obtained by the classifiers. This location determines decision values $f_k(\boldsymbol{x}_i)$ and that of the primary competitor $m_{i,k}$.
(2) The membership function implements a decision scheme combining output scores of the OVA classifiers. The scheme should provide gradual (fuzzy) decision supports, proportional to the difference $(f_k(\boldsymbol{x}_i) - m_{i,k})$. Membership grade for the characteristic point where a tie occurs, $f_k(\boldsymbol{x}_i) = m_{i,k}$, is set to 0.5
(3) When f_k outperforms the primary competitor $(f_k(\boldsymbol{x}_i) > m_{i,k})$, then we must obtain large membership degrees $\mu_k(\boldsymbol{x}_i) \in [0.5, 1]$, while if $f_k(\boldsymbol{x}_i) < m_{i,k}$ the membership grades should be low, taking values in the range $\mu_k(\boldsymbol{x}_i) \in [0, 0.5]$.
(4) Assuming that $m_{i,k} < 1$, a second characteristic point occurs when $f_k^*(\boldsymbol{x}_i) = 1$. In this case, membership degree is set to a threshold $\mu_k(\boldsymbol{x}_i) = \gamma$ (i.e., $\gamma = 0.8$), decided by the user. The parameter γ denotes level of confidence assigned to those patterns lying in the positive margin of classifier f_k. Further, when $m_{i,k} > 1$, the threshold is obtained at greater decision values: $f_k^*(\boldsymbol{x}_i) = 2m_{i,k} - 1$. Threshold determination, along with the decision tie in (2), completes the setting of membership function shape.

For visualization purposes, in Fig. 1 we consider a simple two-dimensional artificial dataset comprising 42 patterns $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}]^T \in \Re^2$, partitioned into three classes. Following the aforementioned procedure, a set of 3 binary K-SVM classifiers is constructed (f_k, $k=1, 2, 3$) to separate class k from the remaining two classes. In this figure, we focus on determining fuzzy degrees $\mu_1(\boldsymbol{x}_i)$, i.e., the fuzzy membership to class c_1, of three distinct patterns $\boldsymbol{x}_P$, $\boldsymbol{x}_Q$, and $\boldsymbol{x}_R$ with $m_{i,k} < 1$. Pattern $\boldsymbol{x}_P$ is a characteristic of those patterns that are well classified as class-1 patterns, lying clearly on the positive side of the boundary defined by f_1. Pattern $\boldsymbol{x}_Q$ refers to ambiguous patterns that are located at overlapping areas designated by boundaries obtained by f_1 and the primary competitor $m_{i,1}$. Finally, pattern $\boldsymbol{x}_R$ refers to patterns that belong to a class different than class 1. The fuzzy membership functions associated with the three patterns are shown in the right side of Fig. 1, with a threshold of $\gamma = 0.8$. Three regions of interest are highlighted in each one (transition zones), namely, regions A–C.

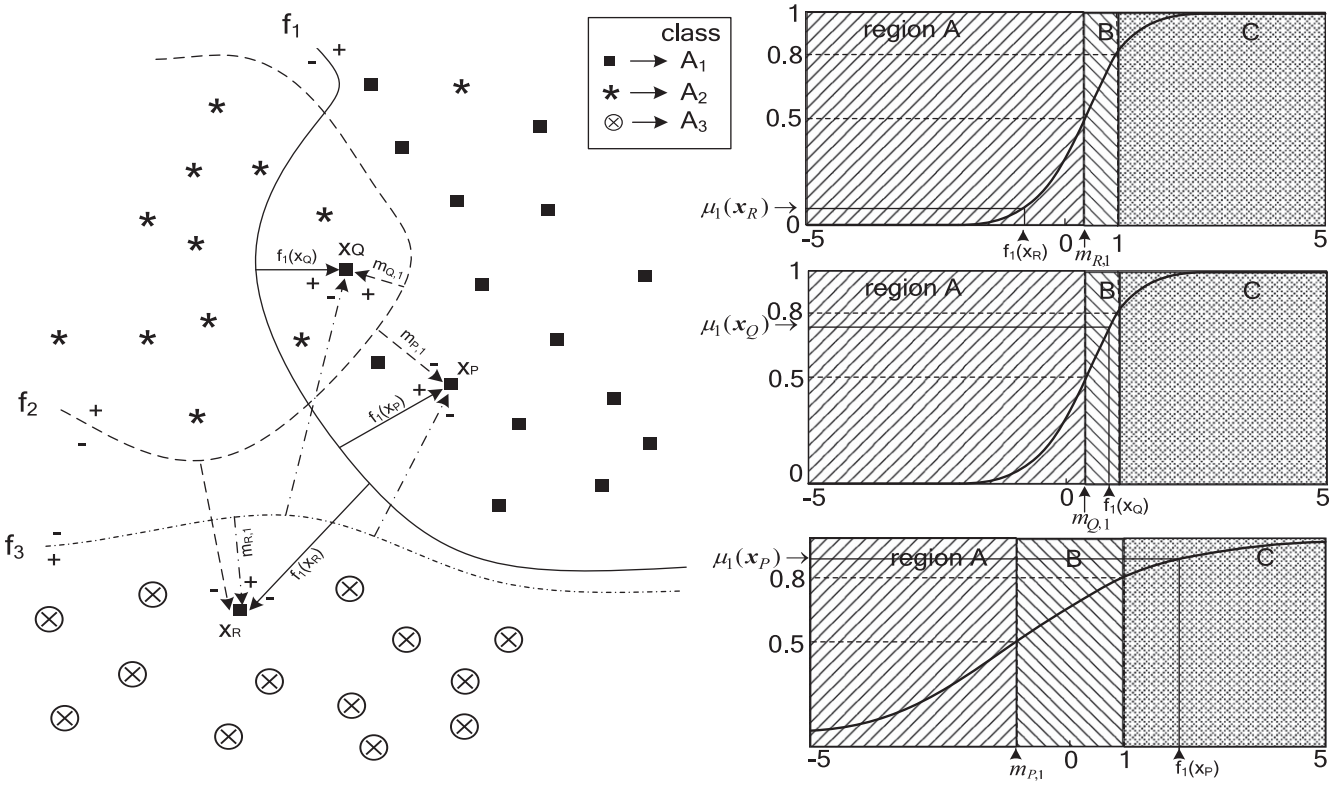

Fig. 1. Fuzzy degree determination $\mu_1(\boldsymbol{x}_i)$, $i \in \{P, Q, R\}$ for patterns $\boldsymbol{x}_P$, $\boldsymbol{x}_Q$, and $\boldsymbol{x}_R$.

1. *Region A (class-k patterns)*: When $f_k(\boldsymbol{x}_i) > 1$, pattern $\boldsymbol{x}_i$ lies beyond the margin on the positive side and hence it can be classified to class k with high confidence. For $f_k(\boldsymbol{x}_i)=1$ (positive margin), the membership grade is set to a threshold γ. This degree tends to 1 when decision value $f_k(\boldsymbol{x}_i)$ increases further. For example, decision function $f_1(\boldsymbol{x}_P)$ (positive value, larger than 1) is significantly bigger than $m_{P,1}=f_2(\boldsymbol{x}_P)$ (negative value), and therefore, $\boldsymbol{x}_P$ receives a membership degree larger than γ.
2. *Region B (Ambiguous patterns)*: If $m_{i,k} < f_k(\boldsymbol{x}_i) \le 1$, the training pattern is assigned to class k and receives membership degrees between 0.5 and γ. Especially for pattern $\boldsymbol{x}_Q$, decision function $f_1(\boldsymbol{x}_Q)$ (positive value) is slightly larger than its competing classifier $m_{Q,1}$ (also positive). Furthermore, since $f_1(\boldsymbol{x}_Q) \le 1$, $\boldsymbol{x}_Q$ is located within the margin of f_1 (on the positive side). Therefore, it can be assigned to class c_1 with a moderate confidence in the range [0.5, γ].
3. *Region C (non-class-k patterns)*: Since $f_k(\boldsymbol{x}_i) < m_{i,\,k}$ in this region, $\boldsymbol{x}_i$ activates to a larger extent the classifier dictated by $m_{i,k}$. In our example, decision value $f_1(\boldsymbol{x}_R)$ (negative value) is lower than $m_{R,1}=f_3(\boldsymbol{x}_R)$ (positive value), indicating that pattern $\boldsymbol{x}_R$ lies in the region of class c_3. On the other hand, if $f_k(\boldsymbol{x}_i)=m_{i,k}$, pattern $\boldsymbol{x}_i$ has the same probability to be classified either correctly or wrongly and hence its fuzzy confidence is set to 0.5.

3. Proposed feature selection method

In this section, we present the proposed FS approach, based on a fuzzy complementary criterion (FuzCoC). The fuzzy partition vector (FPV) is first introduced, used as a local evaluation measure of features, along with some relevant definitions in the fuzzy domain. FPVs contain membership values of each pattern to its own class and their construction is achieved using the aforementioned FO-K-SVM. In the following, we elaborate with the suggested feature selection algorithm (SVM-FuzCoC), leading to discriminatory and non-redundant features.

3.1. Fuzzy partition vector (FPV)

The training patterns in D are initially sorted by class labels:

$$D=\{D_1,\ldots,D_k,\ldots,D_K\} \tag{6}$$

where $D_k=\{\boldsymbol{x}_{i_1},\ldots,\boldsymbol{x}_{i_n},\ldots,\boldsymbol{x}_{i_{N_k}}\}=\{\boldsymbol{x}_{i_n}\,|\,i_n\in\mathcal{A}_k, |\mathcal{A}_k|=N_k\}$ denotes the set of class k patterns, $\mathcal{A}_k$ is the set of indexes of the training examples belonging to class c_k, and $|\cdot|$ stands for cardinality of $\mathcal{A}_k$. N_k is the number of patterns included in D_k, with $\sum_{k=1}^{K} N_k = N$. Notice that class ordering in D is irrelevant, not affecting membership grade determination and feature selection results.

Following the OVA methodology, we initially train a set of M binary K-SVM classifiers on each single feature, to obtain fuzzy membership of each pattern to its class. Let $x_{i,\,j}$ denote the feature j component of pattern $\boldsymbol{x}_i$, $i=1,\ldots,N$. According to FO-K-SVM, fuzzy membership value $\mu_k(x_{i,\,j})\in[0,1]$ of $x_{i,\,j}$ to class k is computed by

$$\mu_k(x_{i,j})=\begin{cases} 0.5 & \text{if } f_k(x_{i,j})=m_{i,j,k}=1 \\ \dfrac{1}{1+e^{\left(\ln\left(\frac{1-\gamma}{\gamma}\right)\right)\cdot\left[\frac{f_k(x_{i,j})-m_{i,j,k}}{|1-m_{i,j,k}|}\right]}} & \text{if } m_{i,j,k}\neq 1 \end{cases} \tag{7}$$

where $f_k(x_{i,j})$ is the decision value of kth K-SVM binary classifier trained by $x_{i,\,j}$ and

$$m_{i,j,k}=\max_{\ell\neq k} f_\ell(x_{i,j}) \tag{8}$$

the maximum decision value obtained by the rest $(k-1)$ K-SVM binary classifiers in $\overline{C}_k$. Concatenating the membership values of class k patterns we form the set

$$\mathcal{M}_{k,j}=\{\mu_k(x_{1,j}),\ldots,\mu_k(x_{i,j}),\ldots,\mu_k(x_{N_k,j})\}=\{\mu_k(x_{i,j})|\,i\in A_k\},\quad k=1,\ldots,M \tag{9}$$

where $\mathcal{M}_{k,j}\in\Re^{N_k}$.

Definition 1. The fuzzy partition vector (FPV) of feature j is defined as

$$G(j)=\{\mu_G(x_{1,j}),\ldots,\mu_G(x_{i,j}),\ldots,\mu_G(x_{N,j})\}=\{\mathcal{M}_{1,j},\ldots,\mathcal{M}_{k,j},\ldots,\mathcal{M}_{M,j}\},\quad G(j)\in\Re^N \tag{10}$$

where

$$\mu_G(x_{i,j})=\mu_{c_i}(x_{i,j})\in[0,1],\quad i=1,\ldots,N \tag{11}$$

Generally, $\mu_G(x_{i,j})$ is determined using the general formula (7) by replacing k with c_i, i.e., the class label that pattern $x_{i,\,j}$ belongs. Each FPV can be regarded as a fuzzy set defined on the patterns universe of discourse D:

$$G(j)=\{(x_{i,j},\mu_G(x_{i,j}))|x_i\in D\},\quad |D|=N,\quad i=1,\ldots,N \tag{12}$$

where $\mu_G(x_{i,\,j})$ denotes the membership value of $x_{i,j}$ to fuzzy set G. From the above definition, FPVs exhibit the following properties:

(1) $G(j)$ represents classification accuracy for every pattern, provided by information contained in feature j. The appearance of high membership degrees in $G(j)$ for a large number of patterns suggests that pattern classes are well separated, as viewed from feature j, indicating that it is a highly discriminating feature.
(2) FPV reveals efficacy of data coverage along patterns. Depending on pattern distribution within the feature space, a particular $G(j)$ exhibits adequate coverage of some patterns while another FPV covers possibly a different pattern subset.
(3) FPV unfolds membership grades along patterns, thereby forming a local evaluation measure with respect to patterns. Hence, $G(j)$ provides a thorough outlook of classification abilities of feature j, which enables a more efficient handling of candidate features

Definition 2. The cardinality of a FPV $G(j)$ is defined as [28]:

$$|G(j)|=\sum_{i=1}^{N}\mu_G(x_{i,j}) \tag{13}$$

$|G(j)|$ defines a quantitative means to evaluate classification ability of feature j. Assuming that training patterns of all classes along feature j are well separated, membership degrees $\mu_G(x_{i,j})$ of the FPV are high (close to 1) for a large portion of patterns. In that case, $|G(j)|$ takes large values (close to N), which implies that discrimination ability of feature j is high. Contrarily, for large overlapping between the classes, cardinality receives lower values (close to N/K) and classification ability of feature j is assumed to be poor.

To highlight the above notions, Fig. 2 depicts a typical $G(j)$ ($\gamma=0.8$) with sufficient data coverage. As can be seen, there are many patterns receiving large membership values (greater than 0.8), which indicates that they are well separated when considering feature j alone. Patterns with FPV values in the range [0.5,0.8] are also well separated using this feature but with a smaller confidence level, as these data points probably lie within the margin of the K-SVM binary classifier, on the positive side. A smaller part of patterns receives lower membership values (less than 0.5), suggesting that feature j is unable to classify them

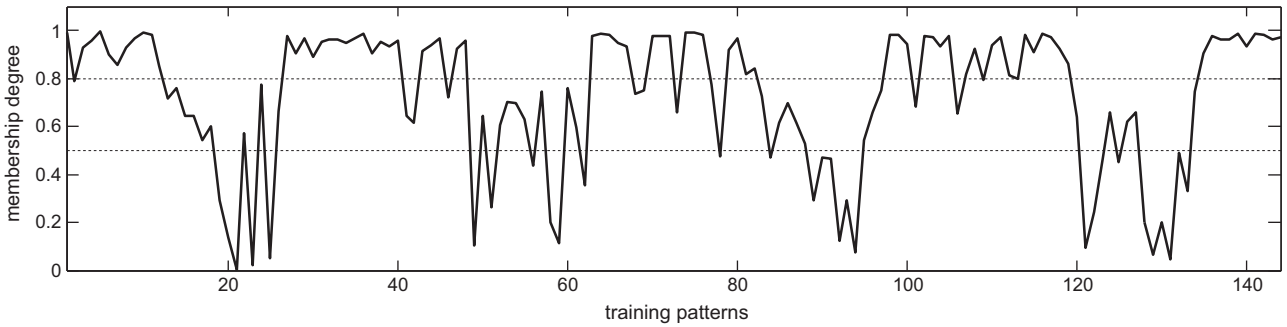

Fig. 2. Typical FPV $G(j)$ associated with feature j.

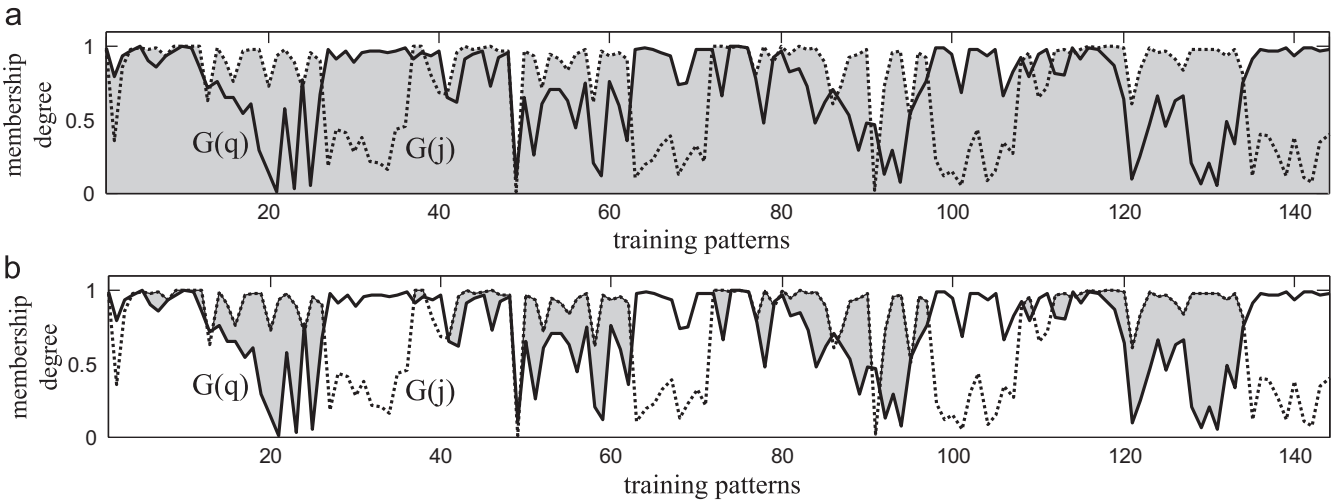

Fig. 3. Union of FPVs $G(j)$and $G(q)$ (grey area in (a)) and their bounded difference (grey area in (b)).

correctly. The above demonstration shows that FPV offers a transparent view of the discrimination quality of each feature, revealing highly classified pattern subsets and those that are not adequately covered by the feature. In contrast, cardinality $|G(j)|$ provides only a global evaluation of features, thereby disregarding their local characteristics along the patterns.

Definition 3. The fuzzy union of two FPVs $G(j)$ and $G(q)$ is defined as

$$G(j) \cup G(q) = \tilde{G}(j,q) \tag{14}$$

where the union is implemented here by the max operator [28]:

$$\mu_{\tilde{G}}(x_i) = \max\{\mu_{G(j)}(x_{i,j}), \mu_{G(q)}(x_{i,q})\}, \quad i = 1,\ldots,N \tag{15}$$

The union FPV, $\tilde{G}(j,q)$, aggregates the classification evidence given by features j and q (grey area in Fig. 3(a)). It exhibits high membership values for patterns that can be correctly classified using either feature j or q. Hence, in our framework, it serves as a means to recognize discrimination ability of different feature combinations.

Definition 4. The bounded difference $G(j)| - |G(q)$ between two FPVs is defined as [28]

$$\mu_{G(j)|-|G(q)}(x_i) = \max\{0, \mu_{G(j)}(x_{i,j}) - \mu_{G(q)}(x_{i,q})\}, \quad i = 1,\ldots,N \tag{16}$$

$G(j)| - |G(q)$ represents the excess of evidence provided by feature j with regard to the one given by feature q (grey area in Fig. 3(b)). Therefore, the bounded difference is employed here as a tool to identify the additional contribution offered by a candidate feature on a set of previously selected features.

3.2. SVM-FuzCoC algorithm

The proposed SVM-FuzCoC is an iterative forward FS method, incorporating the local evaluation measure of FPVs and the FuzCoC principles. Consider a set of initial features, $\mathcal{S} = \{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_j,\ldots,\boldsymbol{z}_n\}$, where $\boldsymbol{z}_j = [x_{1,\,j},\ldots,x_{i,\,j},\ldots,x_{N,\,j}]^T$ and n denotes the total number of features. For each feature $\boldsymbol{z}_j$, we construct in advance the associated FPV by invoking the FO-K-SVM technique: $G(\boldsymbol{z}_j) = \{(x_{i,j},\ \mu_G(x_{i,\,j}))\}$, $i = 1,\ldots,N\}$, $\mu_G(x_{i,j}) = \mu_{c_i,j}(x_i)$, $j = 1,\ldots,n$. Before proceeding, we introduce two important quantities, namely, the cumulative set and the additional contribution, used in the derivation of SVM-FuzCoC.

Definition 5. Let $FS(p) = \{\boldsymbol{z}_{\ell_1},\ldots,\boldsymbol{z}_{\ell_p}\}$ denote the set of p features selected up to and including iteration p. The cumulative set $CS(p)$ is an FPV representing the aggregating effect (union) of FPVs of the features contained in $FS(p)$:

$$CS(p) = G(\boldsymbol{z}_{\ell_1}) \cup \cdots \cup G(\boldsymbol{z}_{\ell_p}) = CS(p-1) \cup G(\boldsymbol{z}_{\ell_p}) \tag{17}$$

$CS(p)$ provides an approximate means to assess quality of data coverage achieved by the features selected at the pth iteration (joint classification).

Definition 6. Assume that $\boldsymbol{z}_{\ell_p}$ is a candidate feature to be selected at iteration p. $AC(p,\ell_p)$ denotes the additional contribution of $\boldsymbol{z}_{\ell_p}$ with respect to the cumulative set $CS(p-1)$ obtained at the preceding iteration, and is determined by

$$AC(p,\boldsymbol{z}_{\ell_p}) = G(\boldsymbol{z}_{\ell_p})| - |CS(p-1) \tag{18}$$

$AC(p,\ell_p)$ represents the surplus of membership grades offered by $G(\boldsymbol{z}_{\ell_p})$, i.e., the FPV associated with feature $\boldsymbol{z}_{\ell_p}$, as compared with the aggregation of previously selected feature FPVs.

Feature selection according to SVM-FuzCoC, proceeds along the following steps.

1. **Initialization**
   1.1. Given the feature set $\mathcal{S}=\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_j,\ldots,\boldsymbol{z}_n\}$, compute the feature FPVs $G(\boldsymbol{z}_j)$, $j=1,\ldots,n$, using (10) and (11)
   1.2. Set $CS(0)=\emptyset$ and $FS(0)=\emptyset$
2. **Select the first feature**: find feature $\boldsymbol{z}_{\ell_1}$ such that

$$\boldsymbol{z}_{\ell_1}=\arg\max_{\ell=1,\ldots,n}\{|G(\boldsymbol{z}_\ell)|\} \tag{19}$$

3. Set $CS(1)=G(\boldsymbol{z}_{\ell_1})$, $FS(1)=FS(0)+\{\boldsymbol{z}_{\ell_1}\}$
4. For iterations $p=2,3,\ldots$ perform the following:
   4.1. FuzCoC feature selection
      4.1.1. Compute the additional contribution of the remaining features

$$AC(p,\boldsymbol{z}_j)=|G(\boldsymbol{z}_j)|-|CS(p-1),\quad j=1,\ldots,n,\quad j\neq\ell_1,\ldots,\ell_{(p-1)}$$

      4.1.2. Find $\boldsymbol{z}_{\ell_p}\in S$ such that

$$\ell_p=\arg\max_{\substack{j=1,\ldots,n\\ j\neq\ell_1,\ldots,\ell_{(p-1)}}}\{|AC(p,\boldsymbol{z}_j)|\} \tag{20}$$

   4.2. Calculate the percentage improvement of $\boldsymbol{z}_{\ell_p}$ with respect to $CS(p-1)$:

$$h_{\ell_p}=\frac{|AC(p,\boldsymbol{z}_{\ell_p})|}{|CS(p-1)|}\times 100\% \tag{21}$$

   4.3. IF $h_{\ell_p}>e_{\boldsymbol{z}}$ THEN
      { 4.3.1 $CS(p)=CS(p-1)\cup G(\boldsymbol{z}_{\ell_p})$
      4.3.2 $FS(p)=FS(p-1)+\{\boldsymbol{z}_{\ell_p}\}$
      4.3.3 Increment $p\leftarrow p+1$ and go to step 4.1}

   ELSE Terminate FuzCoC procedure at iteration m.
5. **Output**: the set $FS(m)=\{\boldsymbol{z}_{\ell_1},\ldots,\boldsymbol{z}_{\ell_m}\}$ of m finally selected features.

3.3. FuzCoC feature selection

Fig. 4 highlights the feature selection principles within the sequential forward selection loop of SVM-FuzCoC. The FPV $G(\boldsymbol{z}_j)$ indicates discrimination capability of $\boldsymbol{z}_j$ to classify individually each one of the training patterns. Interpreted in a different way, $G(\boldsymbol{z}_j)$ represents the fuzzy relevance of $\boldsymbol{z}_j$ to the target classes c, as a fuzzy set along the patterns $G(\boldsymbol{z}_j)=\tilde{D}(c;\boldsymbol{z}_j)$. Further, FPV cardinality $|G(\boldsymbol{z}_j)|$ quantifies in fuzzy terms the class relevance of $\boldsymbol{z}_j$ for all patterns of the dataset and is designated by areas 2 and 3 in Fig. 4.

Initially, SVM-FuzCoC selects the most powerful feature $\boldsymbol{z}_{\ell_1}$ at step 2, i.e., the feature having the highest FPV cardinality $|G(\boldsymbol{z}_\ell)|$. Assume a feature subset $FS(p-1)$ selected prior to the pth iteration. Classification ability of $FS(p-1)$ is described by the cumulative set $CS(p-1)$, while $|CS(p-1)|$ measures the joint relevance of features contained in $FS(p-1)$, designated by areas 1 and 2. The fuzzy intersection between $CS(p-1)$ and $G(\boldsymbol{z}_j)$ is interpreted here as the fuzzy redundancy of $\boldsymbol{z}_j$ in regard to the previously selected features in $FS(p-1)$ denoted by $\tilde{R}(FS(p-1);\boldsymbol{z}_j)$:

$$\tilde{R}(FS(p-1);\boldsymbol{z}_j)=CS(p-1)\cap G(\boldsymbol{z}_j) \tag{22}$$

where the fuzzy intersection is implemented by the min operator [28]. Similarly, $|\tilde{R}(FS(p-1);\boldsymbol{z}_j)|$ measures the overall redundancy (area 2), indicating the common classification evidence shared by $\boldsymbol{z}_j$ and $FS(p-1)$. The fuzzy set $AC(p,\boldsymbol{z}_j)=AC(c;\boldsymbol{z}_j|\,FS(p-1))$, given by (18), denotes additional contribution (relevance to classes) provided by $\boldsymbol{z}_j$ given $FS(p-1)$. In view of (22), it can be shown that $AC(p,\boldsymbol{z}_j)$ can also take a different form embracing $G(\boldsymbol{z}_j)$ and $\tilde{R}(FS(p-1);\boldsymbol{z}_j)$:

$$AC(p,\boldsymbol{z}_j)=|G(\boldsymbol{z}_j)|-|CS(p-1)=G(\boldsymbol{z}_j)|-|\tilde{R}(FS(p-1),\boldsymbol{z}_j) \tag{23}$$

The aggregation of additional contribution and redundancy forms the overall feature relevance to classes:

$$G(\boldsymbol{z}_j)=\tilde{R}(FS(p-1);\boldsymbol{z}_j)\oplus AC(p,\boldsymbol{z}_j) \tag{24}$$

where $A\oplus B$ denotes the bounded sum of fuzzy sets A and B [28], determined by

$$\mu_{A\oplus B}(\boldsymbol{x})=\min\{1,\mu_A(\boldsymbol{x})+\mu_B(\boldsymbol{x})\} \tag{25}$$

Based on (23), cardinality of $AC(p,\boldsymbol{z}_j)$ can now computed as follows:

$$|AC(p,\boldsymbol{z}_j)|=|G(\boldsymbol{z}_j)|-|\tilde{R}(FS(p-1),\boldsymbol{z}_j)|=|\tilde{D}(c;\boldsymbol{z}_j)|-|\tilde{R}(FS(p-1),\boldsymbol{z}_j)| \tag{26}$$

$|AC(p,\boldsymbol{z}_j)|$ quantifies the additional contribution offered by $\boldsymbol{z}_\ell$ and is graphically delineated by area 3 in Fig. 4. The above formula shows that the FuzCoC criterion introduced in this paper takes into consideration both the feature dependence to class targets (relevance) and redundancy, focusing on the difference between the two.

The primary objective characterizing an ideal feature search algorithm is to find the feature that maximizes area 3 [48]. The suggested SVM-FuzCoC succeeds in the above goal incrementally. Starting with $\boldsymbol{z}_{\ell_1}$ selected at step 1, for subsequent iterations $p=2,3,\ldots$ our method selects those features $\boldsymbol{z}_{\ell_p}$ that provide the greatest additional contribution $|AC(p,\boldsymbol{z}_{\ell_p})|$ with respect to the cumulative set $CS(p-1)$ obtained by the existing features in $FS(p-1)$. FuzCoC promotes co-operation between features, ensuring that the newly incoming feature is complementary, namely, it produces higher membership grades for a subset of patterns not adequately covered by the previously selected features.

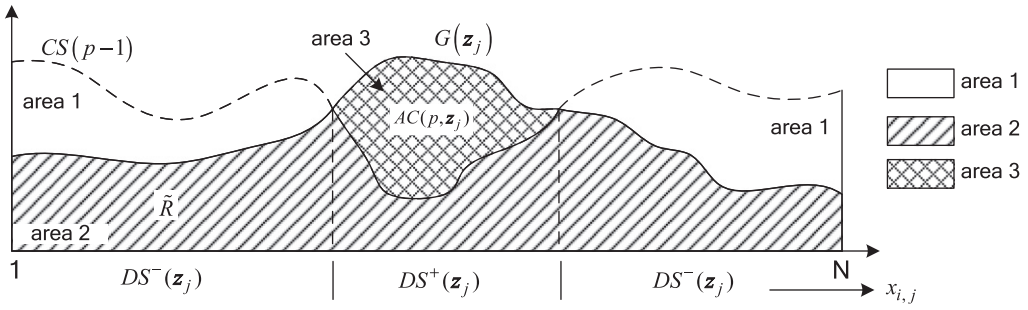

Fig. 4. Illustration of FS principles within the SVM-FuzCoC.

It can be shown that the cumulative set $CS(p)$ obtained by aggregating $CS(p-1)$ and a candidate $G(\boldsymbol{z}_j)$ is derived by

$$CS(p) = CS(p-1) \cup G(\boldsymbol{z}_j) = CS(p-1) \oplus AC(p,\boldsymbol{z}_j) \tag{27}$$

and $CS(p)$ cardinality is written as

$$|CS(p)| = |CS(p-1)| + |AC(p,\boldsymbol{z}_j)| \tag{28}$$

From (28) it can be concluded that maximizing $|AC(p,\boldsymbol{z}_j)|$ implies that the following condition is fulfilled:

$$|CS(p)| = |CS(p-1) \cup G(\boldsymbol{z}_{\ell_p})| = \max_{\boldsymbol{z}_j \in FS(p-1)} |CS(p-1) \cup G(\boldsymbol{z}_j)| \tag{29}$$

This indicates that at each iteration SVM-FuzCoC tries to maximize the aggregating effect ($CS(p)$) of $FS(p) = FS(p-1) + \{\boldsymbol{z}_{\ell_p}\}$. Accordingly, at the end of FS we obtain a set of complementary and non-redundant features with good data coverage ($|CS(m)| \to N$):

$$CS(m) = G(\boldsymbol{z}_{\ell_1}) \cup G(\boldsymbol{z}_{\ell_2}) \cup \cdots \cup G(\boldsymbol{z}_{\ell_m}) \tag{30}$$

Rigorous derivations of some of the aforementioned relations can be found in the appendix. Some recent feature selection techniques, with fruitful results, search for features that maximize a metric $\Phi(\boldsymbol{z}_j)$ defined by

$$\max_{\boldsymbol{z}_j \in FS(p-1)} \{\Phi(\boldsymbol{z}_j) = D(\boldsymbol{z}_j) - R(\boldsymbol{z}_j)\} \tag{31}$$

where $D(\boldsymbol{z}_j)$ and $R(\boldsymbol{z}_j)$ measure the relevance and redundancy, respectively, of $\boldsymbol{z}_j$. Iterative optimization of $\Phi(\boldsymbol{z}_j)$ leads to the so-called minimal-redundancy–maximal-relevance (mRMR) criterion. In the above methods, relevance and redundancy are determined using statistical approaches such as mutual information measure and discriminant analysis statistics [10–12] (see Section 5.2).

Comparing (26) and (31), it can be seen that there is close resemblance between $|AC(p,\boldsymbol{z}_j)|$ and $\Phi(\boldsymbol{z}_j)$. In this respect, FuzCoC actually acts as an mRMR criterion with three distinctions. First, FuzCoC is developed in the fuzzy domain, making use of the FPV notion and derived through fuzzy operators on feature FPVs. Secondly, as opposed to the existing methods, which rely on global discrimination measures $D(\boldsymbol{z}_j)$, $R(\boldsymbol{z}_j)$ (single metric over all patterns), FuzCoC provides both global ($|G(\boldsymbol{z}_j)|$, $|\tilde{R}(FS(p-1),\boldsymbol{z}_j)|$, $|AC(p,\boldsymbol{z}_j)|$) and local feature evaluation with respect to patterns ($G(\boldsymbol{z}_j)$, $\tilde{R}(FS(p-1),\boldsymbol{z}_j)$, $AC(p,\boldsymbol{z}_j)$). This last merit allows us to locate promising pattern areas where a candidate feature performs better with respect to $FS(p-1)$. Finally, the relevance is computed here directly from the fuzzy membership grades obtained by applying FO-K-SVM on feature $\boldsymbol{z}_\ell$. As a result, we exploit the high classification and generalization capabilities of SVMs. In addition, we avoid the increased computational burden encountered in computations of multivariate probability densities (i.e., mutual information) required by other methods.

An additional novelty of SVM-FuzCoC is the quantity h_{ℓ_p} defined in (21), which denotes the percentage improvement of $\boldsymbol{z}_{\ell_p}$ in regard to pre-selected features. The above metric operates in collaboration with a threshold e_z, determined by the designer (i.e., $e_z = 1\,\%$) and serves as a means to decide FS termination in SVM-FuzCoC. If $h_{\ell_p} \geq e_z$, the feature $\boldsymbol{z}_{\ell_p}$ is incorporated in $FS(p-1)$, implying that it contributes significantly to the existing feature set. Contrarily, when $h_{\ell_p} < e_z$, the currently selected feature offers an inferior improvement, and hence, it can be disregarded with no valuable information loss.

Given the feature FPVs, SVM-FuzCoC is a computationally simple search procedure, involving easy fuzzy operations within the forward search iterations. The only cause of complexity lies on computation of feature FPVs via fuzzy K-SVM. Although we train M SVMs for each feature, computational demands are reasonably low, since calculations are confined to one-dimensional (1-D) input vectors. Additionally, the FPVs are derived in advance, i.e., outside the feature search loop.

3.4. SVM-FuzCoC placed in the classifiers combination framework

The SVM-FuzCoC approach introduces a novel variant in the '*data fusion*' literature [29]. Generally, '*data fusion*' comprises two main categories of methods: '*feature fusion*' and '*decision fusion*'. '*Feature fusion*' methods subsume all features into a composite feature vector and then, FS and transformation techniques are applied to reduce dimensionality of the feature set supplying the classifier. In the '*decision fusion*' approach, individual classifiers are developed of different forms (NNs, statistical, etc.). They are constructed using different feature subsets or possibly trained on patterns coming from different data sources. The method then proceeds to combine their decision outputs through fuzzy integration techniques [30]. Decision fusion is performed by fuzzy operators (i.e., fuzzy union, intersection, and averaging), or using more sophisticated aggregation rules such as the fuzzy integral [31] and decision templates [32]. Adaboost is a different methodology within this framework [33]. It consists of combining several weak classifiers, via weighting, to derive the final decision outputs. Traditionally, weak classifiers are simple linear or LDA classifiers with limited accuracy and moderate generalization capabilities. The above method is recently extended to the fuzzy domain, where the weak classifiers take the form of fuzzy rules, extracted through evolutionary learning [34]. A significant property of the decision fusion methods is that the classifiers to be combined should be dissimilar, implying that they commit different classification errors.

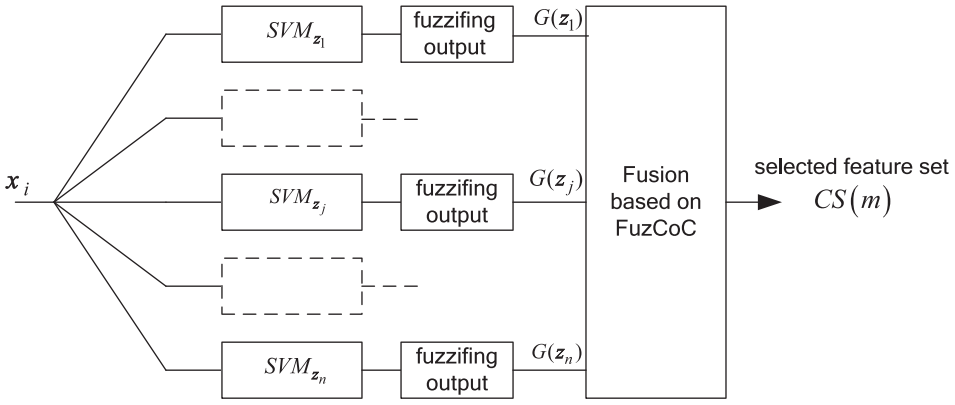

Fig. 5. SVM-FuzCoC interpreted as a decision fusion of weak classifiers.

The proposed SVM-FuzCoC approach can be regarded as a hybrid fusion technique that combines a set of weak fuzzy K-SVM classifiers. An illustrative diagram describing the procedure is shown in Fig. 5. Initially, n fuzzy K-SVM classifiers are trained, one classifier for each feature, to obtain the associated *FPVs*. The *FPVs* embrace, in a pattern-wise base, the decision information (fuzzy decision supports) produced by each one of the n weak fuzzy K-SVM classifiers. Characterization of features as weak classifiers rests on the following rationale. Firstly, each feature has a restricted view of data distribution, depending on how patterns are projected onto the respective feature axis. Hence, for complicated class boundaries, single features are unable to discriminate between classes adequately, therefore leading to lower classification results. Given the candidate FPVs, FuzCoC selects iteratively a subset of co-operative weak classifiers (features), emphasizing on their complementary characteristics along the patterns. The requirement of dissimilarity between the combined classifiers is assured by FuzCoC, since different features perform better on different pattern subsets. For simplicity in computations, decision fusion is achieved here using the fuzzy union operator (max). The cumulative set $CS(m)$ is the aggregation result of decision outputs of the m selected features, exhibiting good data coverage over the entire dataset (maximum cardinality).

4. Illustrative results

4.1. SVM-FuzCoC demonstration

We focus here on the proposed FS method and provide illuminating insights on crucial points of SVM-FuzCoC. Demonstration is carried out on the wine data [35], characterized by three classes and 13 features:$\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_j,\ldots,\boldsymbol{z}_{13}\}$. The original dataset of 178 instances was divided into a training–testing partition (80%–20%, respectively) and SVM-FuzCoC was applied to the training patterns. Evaluation of the method is performed using the k-nearest neighbor classifier with $k=3$ (KNN3).

FS initiates by selecting feature $\boldsymbol{z}_7$ (*Flavanoids*), which maximizes the cardinality criterion $|G(\boldsymbol{z}_j)|$, i.e., having the best global behavior. This feature provides the finest data coverage over the entire training set (Fig. 6(a)). Although it achieves a training performance of 95.83%, its testing accuracy is confined to 64.7%. Most of the patterns are correctly classified with strong

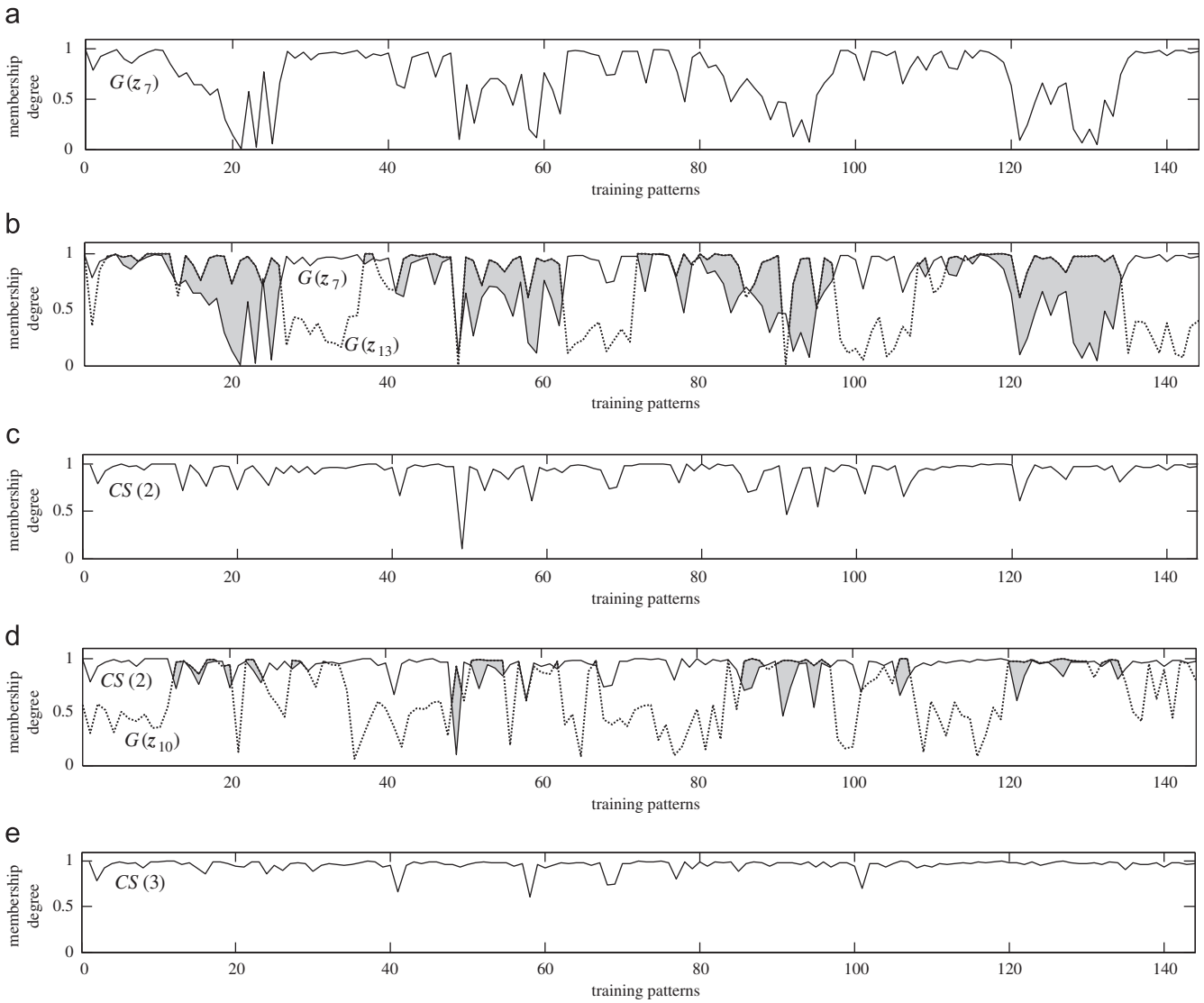

Fig. 6. FS demonstration on wine data: (a) FPV of the first selected feature $\boldsymbol{z}_7$, (b) additional contribution given by $\boldsymbol{z}_{13}$, (c) cumulative set $CS(2)$, (d) additional contribution of $\boldsymbol{z}_{10}$ to $CS(2)$, and (e) final cumulative set $CS(3)$.

confidence (high membership grades). However, there is a portion of patterns that are misclassified when considering $\boldsymbol{z}_7$ alone, hence taking low membership degrees to the desired classes (below 0.5). In the next iteration, FuzCoC selects feature $\boldsymbol{z}_{13}$ (*Proline*), which offers the maximum additional contribution in regard to $\boldsymbol{z}_7$. The newly introduced feature is complementary to $\boldsymbol{z}_7$, in that it exhibits better coverage on certain data subsets that were not adequately covered by $\boldsymbol{z}_7$. The excess of evidence offered by $\boldsymbol{z}_{13}$ is highlighted in Fig. 6(b) (grey area), leading to a percentage improvement of $h_{13}=18.62\%$. The cumulative set $CS(2)$ obtained by aggregating the feature FPVs of $(\boldsymbol{z}_7,\boldsymbol{z}_{13})$ yields an almost perfect coverage of training patterns (Fig. 6(c)), while the training and testing rates are significantly increased to 95.83% and 70.58%, respectively. SVM-FuzCoC concludes by selecting feature $\boldsymbol{z}_{10}$ (*Color intensity*). Despite the fact that $\boldsymbol{z}_{10}$ does not contribute significantly to $CS(2)$ (Fig. 6(d)) and exhibits a moderate value $h_{10}=3.77\%$, it is proved very informative for the testing data. Particularly, the training rate reaches 97.91% while the testing one is considerably enhanced to 91.17%. In addition, the cumulative set $CS(3)$ improved the data coverage further (Fig. 6(e)). FS confines to the subset $(\boldsymbol{z}_7,\boldsymbol{z}_{13},\boldsymbol{z}_{10})$, since the rest of features have h_ℓ values lower than the specified threshold $e_z=2\%$. The effective exploitation of feature FPVs, combined with the FuzCoC principles, assists SVM-FuzCoC to select complementary (non-redundant) features with high discriminatory power.

4.2. Validation of SVM-based membership determination

The fuzzy allocation scheme used for constructing FPVs designates the way the classification capabilities of features are revealed, thus affecting the FS outcome driven by FuzCoC. A synthetic dataset is designed here to demonstrate the efficacy of FO-K-SVM compared with the membership allocation relying on fuzzy C-means (FCM) clustering [36]. Consider a three-dimensional space with 1200 patterns, $\boldsymbol{x}_i=[x_{i,1},x_{i,2},x_{i,3}]^T\in\Re^3$, which are partitioned into three classes of 400 patterns each. Feature components are independent Gaussian variables with probability distributions described by

$$p(x_{i,1})=\begin{cases}\dfrac{1}{\sqrt{2\pi}\sigma}\exp\left(\dfrac{x_{i,1}^2}{2\sigma^2}\right) & \text{if } x_{i,1}\in c_1\\ \dfrac{1}{\sqrt{2\pi}\sigma}\exp\left(\dfrac{(x_{i,1}\pm v_{1,2})^2}{2\sigma^2}\right) & \text{if } x_{i,1}\in c_2\\ \dfrac{1}{\sqrt{2\pi}\sigma}\exp\left(\dfrac{(x_{i,1}\pm v_{1,3})^2}{2\sigma^2}\right) & \text{if } x_{i,1}\in c_3\end{cases},$$

$$p(x_{i,r})=\begin{cases}\dfrac{1}{\sqrt{2\pi}\sigma}\exp\left(\dfrac{(x_{i,r}-v_{r,1})^2}{2\sigma^2}\right) & \text{if } x_{i,r}\in c_1\\ \dfrac{1}{\sqrt{2\pi}\sigma}\exp\left(\dfrac{(x_{i,r}-v_{r,2})^2}{2\sigma^2}\right) & \text{if } x_{i,r}\in c_2,\quad r=2,3\\ \dfrac{1}{\sqrt{2\pi}\sigma}\exp\left(\dfrac{(x_{i,r}-v_{r,3})^2}{2\sigma^2}\right) & \text{if } x_{i,r}\in c_3\end{cases}\tag{32}$$

where the means and standard deviations are $v_{1,2}=8$, $v_{1,3}=15$, $v_{2,1}=2$, $v_{2,2}=4$, $v_{2,3}=6$, $v_{3,1}=7$, $v_{3,2}=8$, $v_{3,3}=9$, and $\sigma=2$.

One random data split is generated, creating a training set of 900 patterns (300 patterns for each class) and a testing set of 300 patterns (the remaining 100 patterns for each class). Fig. 7 shows the scatter plots of the training data for different feature combinations. Using the training data, for each feature $\boldsymbol{z}_j$, $j=1,2,3$, we construct the corresponding FPVs, $G(\boldsymbol{z}_j)$, $j=1,2,3$, via the proposed fuzzy membership determination (SVM-FPVs). For comparison, we also construct a different set of FPVs, $G'(\boldsymbol{z}_j)$, $j=1,2,3$, using the FCM-based membership allocation (FCM-FPVs). According to this method, the membership grade $\mu_{c_i,j}\in[0,1]$ of $x_{i,j}$ to its own class c_i, is determined by applying (only once) the following relation:

$$\mu_{c_i,j}(x_{i,j})=\frac{1}{\sum_{m=1}^{c}[(x_{i,j}-v_{c_i,j})^2/(x_{i,j}-v_{m,j})^2]^{1/b-1}}\tag{33}$$

where

$$v_{c_i,j}=\textstyle\sum_{k\in A_i}x_{i,j}/N_i\tag{34}$$

is the mean of class c_i patterns as projected on feature j, A_i the set of indexes of the training examples belonging to class c_i, N_i the number of class i patterns, and b is a fuzzification factor ($b=2$ here).

The two candidate FPV sets, $SVM-FPV=\{G(\boldsymbol{z}_1),G(\boldsymbol{z}_2),G(\boldsymbol{z}_3)\}$ and $FCM-FPV=\{G'(\boldsymbol{z}_1),G'(\boldsymbol{z}_2),G'(\boldsymbol{z}_3)\}$, are shown in Fig. 8. The membership distributions along the patterns indicate that features are interpreted differently by the two approaches. Apparently, from Fig. 8(a) and (b), feature $\boldsymbol{z}_1$ is the most discriminating, since pattern concentrations along this feature are well separated. This finding is easily verified by $G(\boldsymbol{z}_1)$, which assigns high membership grades for almost every pattern (Fig. 8(a)). Contrarily, $G'(\boldsymbol{z}_1)$ is unable to recognize the high discriminating capabilities of $\boldsymbol{z}_1$, as the projected class means on this feature are very close. This is attributed to the fact that class 2 and 3 patterns are located at disjoint areas, instead of being concentrated in compact class regions. Accordingly, the FCM-based class allocation of (33) apportions low membership degrees (below 0.5) for almost all patterns (Fig. 8(b)). Further, features $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$ are not so descriptive, an observation reflected by the scatter plots and their FPV profiles. Nevertheless, the SVM-FPVs $G(\boldsymbol{z}_2)$ and $G(\boldsymbol{z}_3)$ are able to classify correctly the majority of class 2 and 3 patterns (Fig. 8(c), (e)), in contrast with the FCM-FPVs $G'(\boldsymbol{z}_2)$ and $G'(\boldsymbol{z}_3)$, which show a rather obscure behavior (Fig. 8 (d), (f)).

Next, we apply the FuzCoC-based FS on the two candidate FPV sets using $e_z=2\%$. Effectiveness of the resulting feature subsets is evaluated using a KNN1 classifier. FS operating on SVM-FPVs selects only the most discriminating feature $\boldsymbol{z}_1$ and then terminates, achieving a considerably high testing performance of 98%. On the other hand, the approach operating on FCM-FPVs selects feature $\boldsymbol{z}_2$ as the most informative one, and terminates after selecting feature $\boldsymbol{z}_3$. When presented to the KNN classifier, the obtained subset $\{\boldsymbol{z}_2,\boldsymbol{z}_3\}$ yields a low testing rate of 60.67%. Elaboration with this example demonstrates superiority of the proposed membership determination compared with the FCM-based alternative. Even applied on single features, the high classification and generalization properties of fuzzy K-SVM are retained, rendering it an effective means for computing fuzzy degrees of patterns. The accurate representation of feature capabilities then assists FuzCoC to devise compact sets of complementary features.

4.3. FS termination condition

Quality of features as well as size of the obtained feature subset are usually decided in wrappers by embracing a classifier. However, the classifier's involvement within the search loop aggravates the computational burden. On the other hand, the evaluation metrics used in filter methods do not directly relate to classification ability, thereby yielding inferior recognition rates. To tackle the above issue, SVM-FuzCoC follows a different approach. When regarded in the classifiers combination domain, the cumulative set of FPVs is considered as the integration (fusion) of decision outputs of weak fuzzy K-SVM classifiers

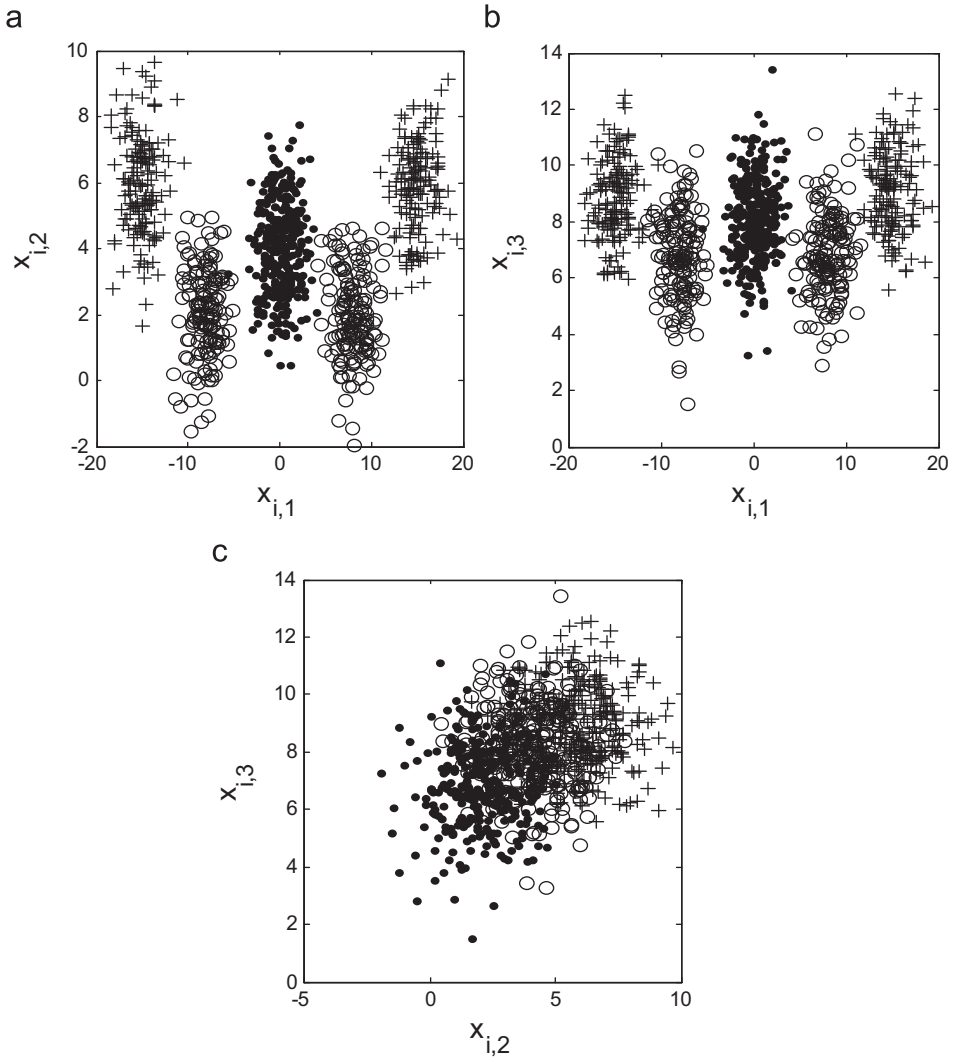

Fig. 7. Scatter plots of the training patterns, projected on feature sets: (a) $\{z_1,z_2\}$, (b) $\{z_1,z_3\}$, and (c) $\{z_2,z_3\}$. The symbols ●, ○, and + indicate patterns of class A_1–A_3, respectively.

emanating from single features. Thus, *CS* cardinality serves, implicitly, as a simple and yet reliable measure to assess the classification ability of the currently selected feature set. Additionally, the percentage improvement h_ℓ quantifying the contribution of newly selected features, combined with a threshold e_z, is used as a criterion for FS termination. In this respect, apart from computational efficiency of the filter approach, SVM-FuzCoC inherits the high accuracy properties encountered in wrappers.

To demonstrate suitability of the termination condition, we apply SVM-FuzCoC on the dermatology problem [35]. A single data partition is performed into training and testing sets (80%–20%) and the resulting features are evaluated via a KNN1 classifier. Fig. 9 shows the variation of h_ℓ along with the testing classification rate versus the number of selected features. It can be seen that h_ℓ decreases exponentially, inversely proportional to the evolution of classification rates. The first incoming features are the most discriminating ones with adequate data coverage and significant contribution on classification rates. Using a threshold $e_z = 1\%$ we obtain six features and the testing performance reaches a threshold of 89.3%. The remaining features with lower h_ℓ values are disregarded, since they offer a marginal improvement in regard to the pre-selected subset.

5. Simulation results

5.1. Real-world datasets

The proposed SVM-FuzCoC is validated on a set of 12 real-world classification problems of varying difficulties. The datasets were categorized into four groups, according to their computational complexity measured by $\dim = Nn$, where N is the total number of patterns and n the initial number of features. The first 9 datasets are taken from UCI repository of machine learning [35], whereas leukemia [37] and diffuse large B-cell lymphoma (DLBCL) [38] were produced from Affymetric gene chips and are available at http://www.genome.wi.mit.edu/cancer/. The data for SRBCT dataset [39] were obtained from cDNA microarrays and are available at http://research.nhgri.nih.gov/microarray/Supplement/. The problem characteristics (number of labeled instances, number of classes, and number of features) are shown in Table 1. In our experiments, we generated 10 random splits, dividing the original datasets into training–testing partitions (70% and 30%, respectively). The training and testing performances were then determined by averaging the respective classification rates over the 10 different random partitions. Comparative analysis is performed in terms of average training and testing performance (%), computational load measured

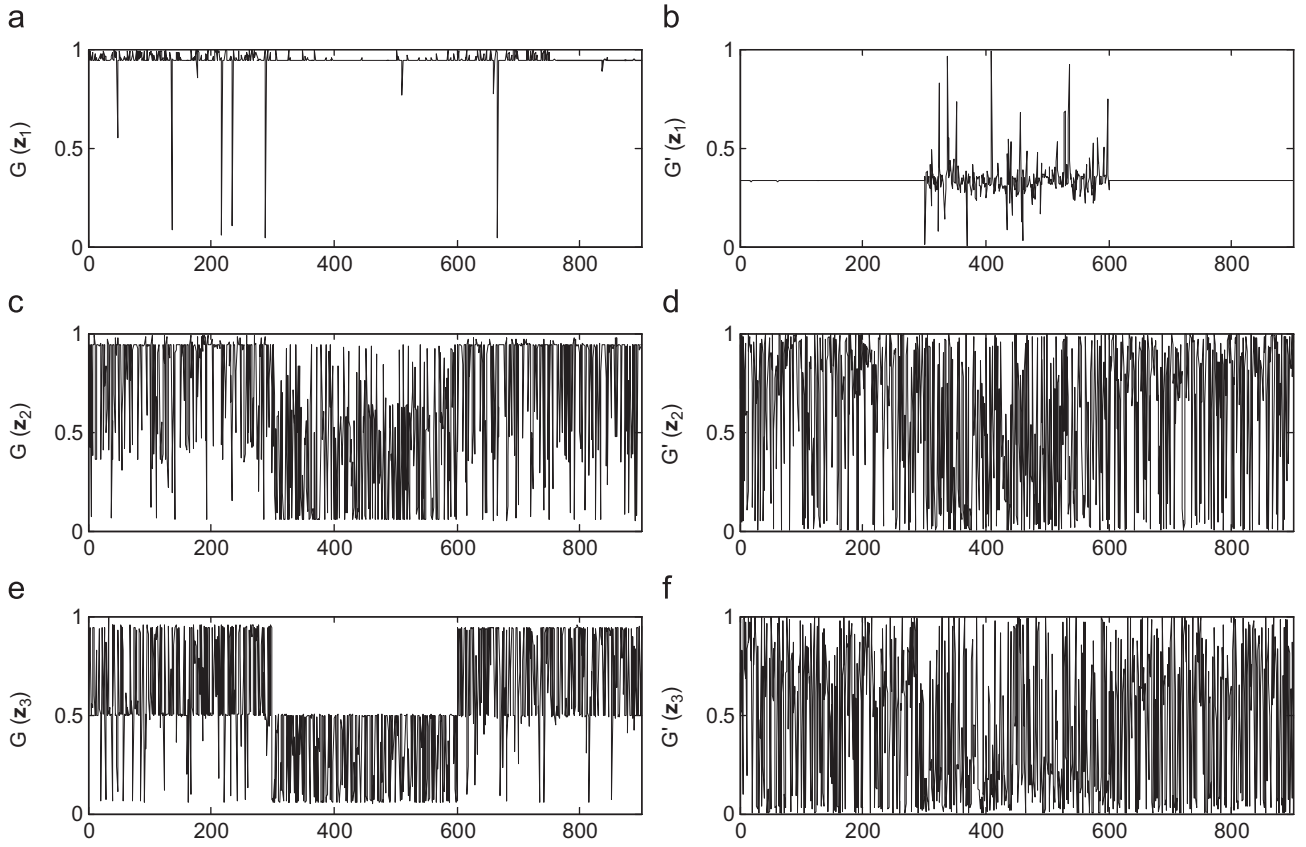

Fig. 8. Proposed SVM – FPVs on synthetic data: (a) $G(\boldsymbol{z}_1)$, (c) $G(\boldsymbol{z}_2)$, (e) $G(\boldsymbol{z}_3)$, and the corresponding FCM – FPVs: (b) $G'(\boldsymbol{z}_1)$, (d) $G'(\boldsymbol{z}_2)$, (f) $G'(\boldsymbol{z}_3)$.

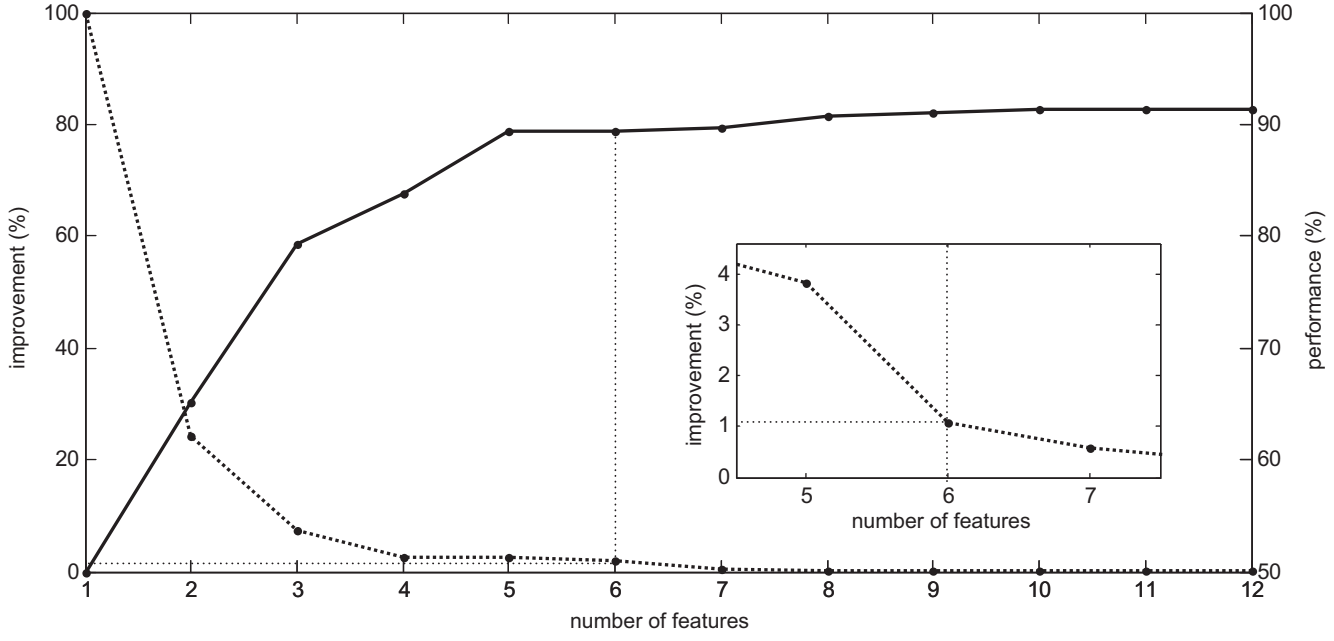

Fig. 9. Evolution of h_ℓ (dashed line—left vertical axis) and testing classification rate of the proposed FS using KNN1 (solid line—right vertical axis) versus number of selected features.

by execution time required for FS termination (in seconds), and dimensionality reduction (DR), defined as

$$DR = \frac{n-\overline{m}}{n} \times 100 \tag{35}$$

where $\overline{m} = \sum_{i=1}^{10} m_i$ denotes the mean value of the number of selected features (m_i) in the 10 random partitions and n the number of original features. The resulting subsets are evaluated via a KNN1 classifier. For each problem, the threshold e_z is suitably chosen from the range [0.1–1%]. Low values of e_z lead to

Table 1
Details of the datasets used in our experiments.

Dataset	Number of patterns	Number of features	Number of classes
Glass	214	9	6
Cleveland	297	13	5
Wine	178	13	3
Ionosphere	351	34	2
Dermatology	366	34	6
Sonar	208	60	2
Wdbc	569	30	2
Page Blocks	5472	10	5
Pen based	10,992	16	10
SRBCT	83	2308	4
Leukemia	72	5147	2
DLBCL	77	7070	2

larger feature subsets (lower DR) and higher testing performance and vice versa.

5.2. Existing feature selection methods

Performance of SVM-FuzCoC is compared to that obtained by eleven well-known FS methods of the literature: SFS, SBS [22], BB [21], SFFS [23], ReliefF [16], Mitra's [18], minimal-redundancy–maximal-relevance (mRMR) [11], MIFS (mutual information feature selection) [10], differential prioritisation (DP) method addressed in [12], the approach proposed in [17] (constraint score-1), and the method suggested in [40] (referred to here as Li's method). Recently developed techniques based on mutual information seem to be a standard, especially in biomedical applications. The mutual information $I(\boldsymbol{z}_j;c)$ is used in these methods to measure the relevance of feature $\boldsymbol{z}_j$ to the target class c. A major issue of this approach lies in the accurate computation of probability density functions in order to estimate mutual information in continuous multi-dimensional feature spaces. In an attempt to overcome the above severe task, research has often focused on calculating pairwise feature evaluations.

When operating in a sequential forward selection fashion, the mRMR method strives to maximize relevance of each feature $I(\boldsymbol{z}_j;c)$ individually, while at the same time minimize redundancy between $\boldsymbol{z}_j$ and the subset of pre-selected features. The feature to be selected at the pth iteration is required to maximize the mRMR criterion:

$$\max_{\boldsymbol{z}_j \in \{\boldsymbol{S}-FS(p-1)\}}\left[I(\boldsymbol{z}_j;c)-\frac{1}{(p-1)}\sum_{\boldsymbol{z}_i \in FS(p-1)} I(\boldsymbol{z}_j;\boldsymbol{z}_i)\right] \tag{36}$$

where $FS(i-1)$denotes the set of already selected features and the set $\{\boldsymbol{F}-FS(i-1)\}$ includes the remaining ones. The density estimation required for computation of mutual information is accomplished using an approach based on direct Parzen-window approximation [11].

The MIFS method requires that, iteratively, the following criterion should be maximized:

$$\max_{\boldsymbol{z}_j \in \{\boldsymbol{S}-FS(p-1)\}}\left[I(\boldsymbol{z}_j;c)-\beta\sum_{\boldsymbol{z}_i \in FS(p-1)} I(\boldsymbol{z}_j;\boldsymbol{z}_i)\right] \tag{37}$$

β is a redundancy parameter used to control the redundancy between features. When $\beta=0$, relevance between features is disregarded and the algorithm selects features solely based on the individual mutual information $I(\boldsymbol{z}_j;c)$. As β increases, relevance between selected features impacts FS and redundancy is reduced. A usual choice compromising between class dependency and feature redundancy is $\beta=0.5$. In the original version of MIFS suggested by Battiti [10], density estimation was based on data histograms, a cumbersome method with serious accuracy and practical limitations. To cope with this shortcoming, we resort to using the Parzen-windows approach in the mutual information computations.

In DP, the original FS problem is actually decomposed into Q two-class sub-problems, which are differentiated through definition of the class labels. OVA and pairwise (PW) approach decomposition methods are employed in DP, generating $Q=K$ and $Q=(K/2)$ binary sub-problems, respectively. A sequential forward selection procedure is applied on each sub-problem q, $q=1,\ldots,Q$, where the following criterion should be maximized:

$$\max_{\boldsymbol{z}_j \in \{\boldsymbol{S}-FS_q(p-1)\}}[(V_{FS_q(p-1)\cup \boldsymbol{z}_i})^a (U_{FS_q(p-1)\cup \boldsymbol{z}_i})^{1-a}] \tag{38}$$

where $V_{FSq(p-1)\cup Z_i}$ measures the level of relevance of the selected feature subset at iteration p of qth sub-problem and $U_{FSq(p-1)\cup Z_i}$ measures the level of antiredundancy in the selected feature subset at iteration p of qth sub-problem. The power factor $a\in(0,1]$ denotes the degree of DP (DDP) between maximizing relevance and antiredundancy. In our experiments OVA decomposition is utilized, where one KNN1 classifier is built for each of the Q selected feature subsets FS_q, $q=1,\ldots Q$. The predicted classes are determined by assigning the class with the strongest decision value.

In [17], the FS algorithm is guided by information in pairwise constraints to find the most relevant feature subsets from the original n features. Constraint Score-1 is computed for each feature, as given by

$$C_r^1=\frac{\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in M}(\boldsymbol{z}_i-\boldsymbol{z}_j)}{\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in C}(\boldsymbol{z}_i-\boldsymbol{z}_j)} \tag{39}$$

where $M=\{(\boldsymbol{x}_i,\boldsymbol{x}_j)|\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class$\}$ denotes the pairwise must-link constraints and $C=\{(\boldsymbol{x}_i,\boldsymbol{x}_j)|\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to different classes$\}$ denotes the pairwise cannot-link constraints. The features are ranked according to their constraint scores in ascending order and the top-ranked features are finally selected, exhibiting the best constraint preserving ability.

Finally, Li's method [40] employs the FCM-based membership allocation scheme given by (24). In the following, the classification capability of features is assessed using a fuzzy set-based measure defined by

$$W(\boldsymbol{z}_j)=\sum_{i=1}^{M}\sum_{k\in A_i}\mu_{c_i,j}(x_{i,j}) \tag{40}$$

Features are first ranked in descending order of $W(\boldsymbol{z}_j)$. A subset of top-ranked features is then retained, which provides acceptable recognition rates for a given classification model.

5.3. SVM-FuzCoC variants

In Section 2.2, we proposed the FO-K-SVM approach for determining membership grades of patterns, used for FPV construction and subsequent feature selection by FuzCoC. Nevertheless, it should be noticed that SVM-FuzCoC does not adhere solely to the above method. Contrarily, any other method might be equivalently employed to achieve the grade evaluation task, relying either on probabilistic or fuzzy principles. The basic requirement is to derive appropriate soft degrees from data, reflecting the classification process and the classification capabilities of features. To this end, apart from FO-K-SVM, we introduce in our experiments several variants of SVM-FuzCoC by considering seven additional membership degree determination methods. Among the probabilistic methods used to convert the SVM outputs and produce calibrated posterior probabilities, the Platt

and the MSVMO methods are parametric whereas Binning, DPS, and Isotonic regression are non-parametric methods.

Crisp assignments: Patterns are assigned to a single class (crisp degrees), following the max principle of OVA scheme as obtained by (3).

FCM-based fuzzy degrees: According to this method, the membership degrees are determined using (24).

Sigmoid fit (Platt): This is a technique for probability calibration especially designed for SVM classifiers [41]. In this case, training examples are used to fit a sigmoid function. The calibrated posterior probability estimates are then obtained as

$$P(k|\boldsymbol{x}_i)=\frac{1}{1+\exp(Af_k(\boldsymbol{x}_i)+B)} \tag{41}$$

The parameters A and B are learned using maximum likelihood estimation.

Sigmoid fit (MSVMO): An alternative way of interpreting and modifying the SVM outputs is also examined utilizing analytical geometry [42]. Posterior probabilities are computed by the appropriate use of a logistic sigmoid on distances $d(x_i)$ from the decision hyperplane:

$$P(k|\boldsymbol{x}_i)=\frac{1}{1+\exp(\tau\, d(\boldsymbol{x}_i))} \tag{42}$$

where the parameter τ is estimated with the maximum likelihood method beforehand.

Binning method: The discrete non-parametric binning method is a histogram technique, used to convert decision outputs of SVM classifiers into well-calibrated posterior probability densities [43,44]. For each feature, we initially construct a set of M classifiers, f_k, $k=1,\ldots,M$, each one separating class k from the rest of classes in $\overline{C}_k$. Based on the decision values $f_k(x_i)$, the binning method is used to provide a calibrated estimate of the class-k posterior probability $P(k|x_i)$ from data. The method proceeds by first sorting training examples according to their decision values, and then dividing them into b subsets of equal size, called bins. The number of bins is decided experimentally and must be small ($b=10$ in our simulations) in order to reduce variance of the binned probability estimates. Given a testing pattern $\boldsymbol{x}_i$, it is placed in a bin according to its score $f_k(\boldsymbol{x}_i)$. The corresponding probability estimate $P(k|\boldsymbol{x}_i)$ is then determined as the fraction of positive training examples that fall within the bin.

DPS method: This is a non-parametric technique [45] for derivation of probability values by recording frequency of an event of interest over many trials. A set of similar patterns $\overline{X}_i$ is generated for each pattern $\boldsymbol{x}_i$ comprising its neighbors. The class-k posterior probability $P(k|\boldsymbol{x}_i)$ is estimated by counting the times that class k is the top choice during $|\overline{X}_i|$ trials, where $|\overline{X}_i|$ denotes the number of patterns contained in $\overline{X}_i$. Neighborhoods of patterns are created for each class by dividing the range of scores into zones and allowing each sample to fall into a zone.

PAV method (isotonic regression): Isotonic regression [46] is a strictly non-parametric method that produces accurate probability estimates bypassing the sigmoidal assumption (variants 3 and 4) and the problem of determining the best number of bins in variant 5 or zones in variant 6. The commonly used PAV algorithm (pair-adjacent violators) is employed to find the stepwise-constant isotonic function that best fits the data according to a mean-squared error criterion. Isotonic regression via PAV may be interpreted as a binning method that properly selects the bins through data.

It should be noticed that the posterior probabilities $P(k|\boldsymbol{x}_i)$ obtained by the methods (3)–(7) are considered as possibilities and are subsequently used for the FPV generation in (10).

5.4. Threshold effect

In this section we investigate the effect of parameter γ involved in (7) on the behavior of SVM-FuzCoC. Considering three classification problems (Dermatology, Wine, and Sonar), Fig. 10 shows classification rates and DR for varying values of γ in the range [0.6,1.0] using the KNN1 classifier. It can be seen that for larger values of γ, FO-K-SVM puts greater membership degrees to patterns, and hence FuzCoC terminates with fewer features (larger DR) required for sufficient data covering (CS cardinality). However, due to the overestimation of fuzzy degrees, a portion of patterns is misclassified (especially those lying within the positive margin), which leads to lower classification rates. Contrarily, for lower values of γ FO-K-SVM assigns smaller degrees to the border patterns, and FuzCoC concludes with a larger feature subset (smaller DR). In addition, the existence of more features combined with more conservative definition of fuzzy grades produces adequately high recognition rates. A reasonable recommendation used in the subsequent simulations is to select $\gamma=0.8$, which provides a good balance between sufficient DR and performance. An alternative option would be to examine results on a grid of different values of γ and obtain an optimal choice according to the designer's preferences.

5.5. Comparative analysis

Tables 2–5 host average results of SVM-FuzCoC compared with the ones obtained by the 11 existing FS methods described in Section 5.2. At this stage, our approach is implemented using the FO-K-SVM fuzzy degree determination method. For ease in the comparisons, the number of features finally selected by mRMR, MIFS ($\beta=0.5$), and Li's method is the same as the one decided by SVM-FuzCoC, i.e., the above methods share the same DR. Implementation of SFS, SBS, SFFS, and BB for the datasets of Table 5 was infeasible, since their time requirements increase rapidly with respect to number of features.

A general finding drawn from the above tables is that SVM-FuzCoC provides the best trade-off between testing accuracy and DR. Particularly, in most cases (8 out of 12 datasets), the proposed method yields the highest testing rate, whereas in four cases (Dermatology, Sonar, WDBC, and SRBCT) SVM-FuzCoC achieves both the highest testing accuracy and the greatest DR simultaneously. Furthermore, in Wine and Ionosphere datasets, our approach gets the greatest DR, with the second and third high classification performance, respectively. Moreover, SVM-FuzCoC accomplishes the second high testing accuracy and DR in Leukemia dataset, whereas in Pen-based data the greatest DR is achieved, though without sacrificing classification performance. As regards computational load, SVM-FuzCoC requires reasonably low execution times comparable with those of Mitra's, mRMR, MIFS, ConScore, and Li's, whereas it considerably outperforms SFS, SBS, SFFS, BB, and DDP-OVA. Concluding, our method is adequately fast, reaching consistently high classification rates in all the problems examined.

5.6. Evaluation of SVM-FuzCoC variants

Tables 6–9 show average results for eight SVM-FuzCoC variants, whereby different degree determination methods are examined, namely, the FO-K-SVM and seven other approaches discussed in Section 5.3. Evaluation on different datasets is performed in terms of testing accuracy, DR, and execution times, using the KNN1 classifier. By definition, the crisp assignments obtained by OVA disregard the information of pattern relative

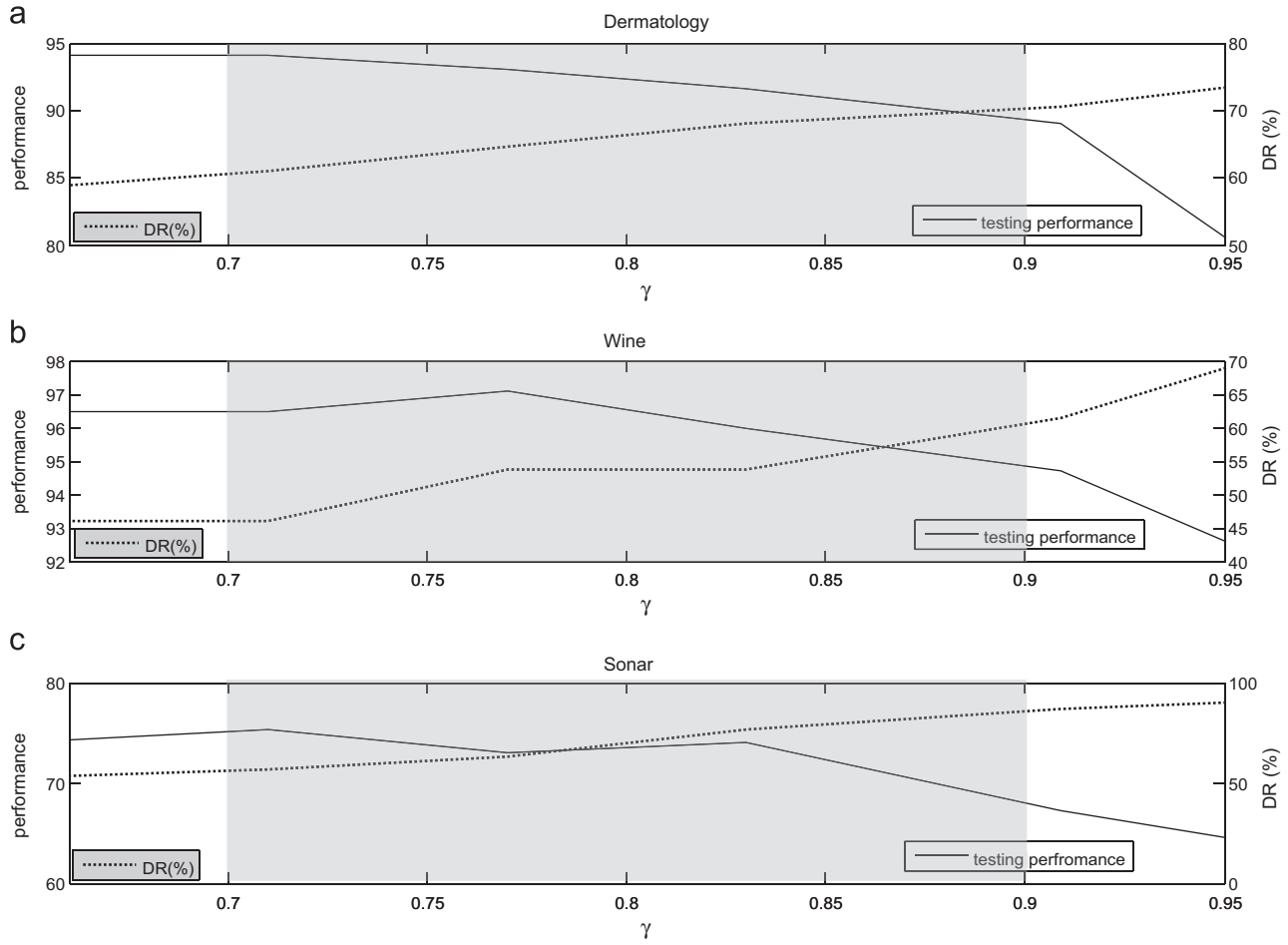

Fig. 10. Evolution of classification rates (left axis) along with DR (%, right axis) with respect to values of γ for three classification problems.

Table 2
Comparison of average results over datasets with dim < 5000.

Dataset :	Glass			Cleveland			Wine		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
SFS	72.24	26.66	0.66	51.79	47.7	1.74	**97.69**	35.38	1.09
SBS	71.77	**37.77**	0.64	54.8	38.5	1.7	94.77	46.15	1.07
SFFS	71.77	**37.77**	1.37	49.55	**53.8**	6.68	96.56	36.92	3.66
BB	70.99	**37.77**	0.31	48.51	**53.8**	0.87	96.53	36.92	0.35
ReliefF	72.76	26.66	0.09	51.47	46.1	0.19	94.29	29.23	0.07
Mitra	70.1	22.22	0.02	45.81	50.8	0.03	95.47	33.84	0.03
Li	72.9	31.1	0.6	49.15	40	0.5	95.96	50.76	0.21
mRMR	70.2	33.33	0.011	48.5	46.1	0.012	97.12	**53.84**	0.012
MIFS	68.7	33.33	0.01	48.49	46.1	**0.011**	94.93	**53.84**	0.01
DDP-OVA	64.05	22.22	0.48	52.18	33.84	0.94	95.44	21.53	0.41
ConScore	66.80	33.33	**0.008**	52.47	46.1	0.013	96.52	46.15	**0.008**
SVM-FuzCoC	**73.36**	33.33	0.6	**61.01**	46.1	0.52	97.12	**53.84**	0.21

distances to competing boundaries. Consequently, ambiguous pattern regions are improperly characterized by this method, resulting in high DR values but with considerably inferior testing accuracies.

The proposed FO-K-SVM provides a good compromise, producing high classification rates while at the same time achieving high DR scores at low computational costs. Particularly, the proposed method accomplishes the highest testing rates over 6 out of the 12 datasets (Glass, Wine, WDBC, and the datasets of high dimensionality SRBCT, Leukemia, and DLBCL). DPS and Binning seem to be attractive alternatives, exhibiting the best performance with regard to testing accuracy over three and two datasets, respectively. MSVMO and Platt's method are less effective, requiring significantly higher execution times due to parameter fitting. The PAV method is more powerful over Pen-based dataset, where there is sufficient number of patterns to prevent overfitting. This finding is compatible with the work of Niculescu-Mizil and Caruana [47], where a learning curve analysis shows that isotonic regression is prone to overfitting when data are scarce. Finally, owing to the implications discussed

Table 3
Comparison of average results over datasets with $5000 < \text{dim} < 15{,}000$.

Dataset :	Ionosphere			Dermatology			Sonar		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
SFS	87.75	65.88	12.7	94.02	44.7	15.04	66.43	61.33	23.5
SBS	84.61	77.64	13.16	91.78	58.23	15.49	62.2	45.33	24.1
SFFS	88.32	75.29	61.01	93.7	62.35	49.82	64.55	61.33	115.3
BB	86.88	75.29	7.08	91.47	62.35	420.7	61.19	61.33	17.9
ReliefF	88.61	72.35	0.33	84.19	59.41	0.37	59.31	59.66	0.17
Mitra	88.04	66.47	0.22	81.91	57.64	0.22	65.53	57.33	0.65
Li	89.03	85.29	1.3	82.47	55.29	1.98	69.48	51	1.65
mRMR	89.74	**88.23**	0.13	92.09	**64.7**	0.1	70.14	**68.33**	0.21
MIFS	86.23	**88.23**	0.12	93.72	**64.7**	0.1	64.61	**68.33**	0.19
DDP-OVA	**90.31**	62.94	3.85	89.05	44.7	11.63	64.00	66.33	18.28
ConScore	89.46	73.52	**0.036**	83.61	**64.7**	**0.039**	59.67	60.00	**0.025**
SVM-FuzCoC	89.46	**88.23**	1.32	**94.11**	**64.7**	2.01	**73.17**	**68.33**	1.68

Table 4
Comparison of average results over datasets with $15{,}000 < \text{dim} < 180{,}000$.

Dataset :	WDBC			Page Blocks			Pen based		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
SFS	94.55	54.83	23.76	94.55	22	239.5	97.42	0	6491
SBS	96.12	68.38	24.32	94.57	22	245.7	96.71	6.25	6289
SFFS	95.59	50.32	101.9	94.3	**32**	773.1	96.71	6.25	26,136
BB	93.67	50.32	51.9	94.39	**32**	164.6	97.1	6.25	1551
ReliefF	95.41	47.09	0.81	94.79	22	317.3	97.34	6.25	18,706
Mitra	92.79	54.83	0.19	94.77	30	**0.07**	96.99	12.5	6.3
Li	93.64	68.38	2.7	94.18	16	57.01	97.1	**25**	802.3
mRMR	95.19	**74.19**	0.1	94.66	30	0.1	97.4	**25**	3.2
MIFS	95.78	**74.19**	0.09	94.66	30	0.09	96.9	**25**	**3.0**
DDP-OVA	95.26	65.80	3.32	94.31	14	417.2	97.42	12.5	1765
ConScore	93.83	67.74	**0.08**	94.83	30	2.17	**98.16**	**25**	13.76
SVM-FuzCoC	**96.48**	**74.19**	2.8	**95.04**	30	57.51	97.22	**25**	812.5

Table 5
Comparison of average results over datasets of high dimensionality with $\text{dim} > 180{,}000$.

Dataset :	SRBCT			Leukemia			DLBCL		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
ReliefF	95.19	97.56	1.01	92.95	98.66	1.90	76.90	99.43	2.80
Mitra	75.55	94.43	814.1	62.23	97.74	1141.2	71.66	98.84	1699.3
Li	83.10	97.83	3.69	72.00	99.02	12.4	80.55	99.29	21.56
mRMR	97.71	98.26	68.69	**95.80**	99.22	85.46	89.83	99.43	92.83
MIFS	87.88	98.26	67.53	90.09	99.22	74.14	92.5	99.43	84.11
DDP-OVA	92.79	98.19	209.8	88.76	**99.81**	227.5	87.00	**99.85**	315.2
ConScore	97.77	98.26	**0.77**	90.19	99.22	**0.93**	74.21	99.29	**1.24**
SVM-FuzCoC	**98.88**	**98.57**	51.37	95.71	99.75	69.02	**93.22**	99.78	91.49

Table 6
Average results over datasets with $\text{dim} < 5000$ using different degree determination methods.

Dataset :	Glass			Cleveland			Wine		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
Crisp	72.58	33.33	0.61	42.15	**84.61**	0.54	95.35	61.53	0.209
FCM	68.67	2.22	**0.005**	49.83	0	**0.008**	96.12	10.76	**0.0048**
Platt	71.1	15.55	14.13	48.57	38.46	22.08	94.25	46.15	7.95
MSVMO	**73.36**	28.88	12.61	41.00	70.76	21.68	88.49	**63.07**	6.30
Binning	72.51	33.33	0.64	51.44	20	0.6	93.66	52.3	0.29
DPS	71.00	**48.88**	3.19	**64.40**	67.69	3.35	**97.12**	53.84	0.95
PAV	71.00	17.77	1.88	42.99	64.61	2.78	94.26	52.30	0.96
FO-K-SVM	**73.36**	33.33	0.6	61.01	46.1	0.52	**97.12**	53.84	0.21

Table 7
Average results over datasets with $5000 < \dim < 15{,}000$ using different degree determination methods.

Dataset :	Ionosphere			Dermatology			Sonar		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
Crisp	91.17	**88.23**	1.39	83.32	**81.17**	2.05	63.95	**93.33**	1.75
FCM	92.85	82.35	**0.017**	93.16	42.35	**0.03**	64.1	53	**0.029**
Platt	89.47	85.29	29.71	86.05	64.11	89.73	**73.42**	61.66	17.92
MSVMO	82.00	**88.23**	19.03	84.00	75.88	81.64	58.63	91.66	18.44
Binning	88.32	85.29	1.49	**96.09**	44.11	2.4	62.05	68	1.81
DPS	**93.17**	77.64	7.35	74.64	77.05	29.87	58.54	49	6.54
PAV	87.46	82.94	4.26	92.45	78.23	9.89	64.15	73.66	4.63
FO-K-SVM	89.46	**88.23**	1.32	94.11	64.7	2.01	73.17	68.33	1.68

Table 8
Average results over datasets with $15{,}000 < \dim < 180{,}000$ using different degree determination methods.

Dataset :	WDBC			Page blocks			Pen based		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
Crisp	91.7	**86.45**	2.88	94.2	30	59.65	88.71	**58.75**	783.1
FCM	92.03	34.19	**0.038**	94.66	0	**1.154**	97.42	0	**5.26**
Platt	92.59	60.64	32.24	95.01	14	404.9	97.12	28.7	3266.4
MSVMO	94.22	84.51	32.29	95.31	**54**	406.8	97.12	22.2	3356.2
Binning	94.03	63.87	3.42	**95.82**	22	67.94	97.42	0	899
DPS	96.28	27.74	23.69	93.75	42	315.2	97.12	22.2	14,474
PAV	89.38	79.35	7.57	94.22	14	194.2	**99.15**	3.75	3331.2
FO-K-SVM	**96.48**	74.19	2.8	95.04	30	57.51	97.22	25	812.5

Table 9
Average results over datasets with $\dim > 180{,}000$ using different degree determination methods.

Dataset	SRBCT			Leukemia			DLBCL		
	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)	Testing accuracy (%)	DR (%)	Time (s)
Crisp	85.15	**99.74**	31.42	84.76	**99.94**	52.30	80.48	**99.95**	89.07
FCM	76.46	99.15	**2.99**	83.04	99.86	**9.56**	77.01	99.18	**18.70**
Platt	97.63	99.18	1747.2	92.95	99.70	3765.4	89.58	99.57	5509.7
MSVMO	82.45	98.13	1574.5	85.98	99.18	3493.6	78.09	99.14	5664.8
Binning	95.29	99.54	40.22	88.14	99.21	76.90	79.05	98.90	139.69
DPS	75.29	98.34	111.26	81.38	99.68	134.15	77.21	99.61	229.84
PAV	97.02	98.51	103.3	92.05	99.74	99.81	88.48	99.42	179.46
FO-K-SVM	**98.88**	98.57	51.37	**95.71**	99.75	69.02	**93.22**	99.78	91.49

in Section 4.2, the FCM-based method succeeds smaller DR values, usually with lower testing rates.

6. Conclusions

A novel feature selection method is proposed in this paper that accomplishes high performance accuracy in adequately small computation time even in high-dimensional datasets. The method relies on a local evaluation criterion assigning class membership degrees along patterns for each feature. For this purpose a fuzzy K-SVM technique is employed due to its generalization properties and computational efficiency in high-dimensional problems. Based on the principles of a fuzzy complementary criterion (FuzCoC), the proposed FS handles simultaneously both discrimination power and complementary characteristics between the features, selecting at each iteration features with the maximum additional contribution in respect to the previously selected features. The superiority of SVM-FuzCoC is verified in terms of testing accuracy, dimensional reduction, and computational load in comparison with other FS methods of the literature.

Appendix

A.1. Proof of equation (23)

Given $CS(p-1)$ at the pth iteration and a candidate feature FPV $G(z_j)$ we define the sets

$$DS^{+}(\boldsymbol{z}_j) = \{x_{i,j} \,|\, \mu_G(x_{i,j}) > \mu_{CS(p-1)}(x_{i,j}), \quad i = 1,\ldots,N\} \tag{A.1}$$

$$DS^{-}(\boldsymbol{z}_j) = D - DS^{+}(\boldsymbol{z}_j) \tag{A.2}$$

For $x_{i,j} \in DS^{-}(\boldsymbol{z}_j)$ we have $\mu_G(x_{i,j}) \le \mu_{CS(p-1)}(x_{i,j})$ and hence

$$\mu_{G(\boldsymbol{z}_j)|-|CS(p-1)}(x_{i,j}) = \max\{0, \mu_G(x_{i,j}) - \mu_{CS(p-1)}(x_{i,j})\} = 0 \tag{A.3}$$

For $x_{i,j} \in DS^{+}(\boldsymbol{z}_j)$ we have

$$\mu_R(x_{i,j}) = \min(\mu_{G(\boldsymbol{z}_j)}(x_{i,j}), \quad \mu_{CS(p-1)}(x_{i,j})) = \mu_{CS(p-1)}(x_{i,j}) \tag{A.4}$$

and so

$$\mu_{G(\boldsymbol{z}_j)|-|CS(p-1)}(x_{i,j}) = \max\{0, \mu_G(x_{i,j}) - \mu_{\tilde{R}(\boldsymbol{z}_j)}(x_{i,j})\} = \mu_{G(\boldsymbol{z}_j)|-|\tilde{R}(\boldsymbol{z}_j)}(x_{i,j}) \tag{A.5}$$

which completes the proof of (23)

A.2. Proof of equation (24)

The right hand of (24) is written as

$$\mu_{\tilde{R}(\boldsymbol{z}_j)\oplus AC(p,\boldsymbol{z}_j)} = \min[1,\min\{\mu_{G(j)}(x_{i,j}),\mu_{CS(p-1)}(x_{i,j})\} + \max\{0,\mu_{G(j)}(x_{i,j})-\mu_{CS(p-1)}(x_{i,j})\}] \tag{A.6}$$

For $x_{i,j}\in DS^{+}(z_j)$ and by definitions (A.1) and (A.2) we have

$$\mu_{\tilde{R}(\boldsymbol{z}_j)\oplus AC(p,\boldsymbol{z}_j)} = \min[1,\mu_{CS(p-1)}(x_{i,j})+\mu_{G(j)}(x_{i,j})-\mu_{CS(p-1)}(x_{i,j})] = \mu_{G(j)}(x_{i,j}) \tag{A.7}$$

In addition, for $x_{i,j}\in DS^{-}(\boldsymbol{z}_j)$ we have

$$\mu_{\tilde{R}(\boldsymbol{z}_j)\oplus AC(p,\boldsymbol{z}_j)} = \min[1,\mu_{G(\boldsymbol{z}_j)}(x_{i,j})+0] = \mu_{G(j)}(x_{i,j}) \tag{A.8}$$

Hence

$$\mu_{\tilde{R}(\boldsymbol{z}_j)\oplus AC(p,\boldsymbol{z}_j)} = \mu_{G(j)}(x_{i,j}) \tag{A.9}$$

A.3. Proof of equation (26):

By definition (13) we have

$$|G(\boldsymbol{z}_j)| = \sum_{x_{i,j}\in DS^{+}(\boldsymbol{z}_j)} \mu_G(x_{i,j}) + \sum_{x_{i,j}\in DS^{-}(\boldsymbol{z}_j)} \mu_G(x_{i,j}) \tag{A.10}$$

Due to (A.1) and (A.2) the following relation holds:

$$\begin{aligned}|\tilde{R}(\boldsymbol{z}_j)| &= \textstyle\sum_{x_{i,j}\in DS^{+}(\boldsymbol{z}_j)}\mu_{\tilde{R}}(x_{i,j})+\sum_{x_{i,j}\in DS^{-}(\boldsymbol{z}_j)}\mu_{\tilde{R}}(x_{i,j})\\ &= \textstyle\sum_{x_{i,j}\in DS^{+}(\boldsymbol{z}_j)}\mu_{CS(p-1)}(x_{i,j})+\sum_{x_{i,j}\in DS^{-}(\boldsymbol{z}_j)}\mu_G(x_{i,j})\end{aligned} \tag{A.11}$$

Subtracting (A.6) and (A.7) we have

$$|G(\boldsymbol{z}_j)|-|\tilde{R}(\boldsymbol{z}_j)| = \sum_{x_{i,j}\in DS^{+}(\boldsymbol{z}_j)} \{\mu_G(x_{i,j})-\mu_{CS(p-1)}(x_{i,j})\} = |AC(p,\boldsymbol{z}_j)| \tag{A.12}$$

which completes the proof of (26).

A.4. Proof of equation (28)

From the right hand of (28) we have

$$\mu_{CS(p-1)\oplus AC(p,\boldsymbol{z}_j)}(x_{i,j}) = \min\{1,\mu_{CS(p-1)}(x_{i,j})+\mu_{AC(p,\boldsymbol{z}_j)}(x_{i,j})\} \tag{A.13}$$

For $x_{i,j}\in DS^{-}(\boldsymbol{z}_j)$, using (23)

$$\mu_{AC(p,\boldsymbol{z}_j)}(x_{i,j}) = \max\{0,\mu_{G(\boldsymbol{z}_j)}(x_{i,j})-\mu_{CS(p-1)}(x_{i,j})\} = 0 \tag{A.14}$$

From (A.14), (A.13) becomes

$$\begin{aligned}\mu_{CS(p-1)\oplus AC(p,\boldsymbol{z}_j)}(x_{i,j}) &= \min\{1,\mu_{CS(p-1)}(x_{i,j})\} = \mu_{CS(p-1)}(x_{i,j})\\ &= \max\{\mu_{G(\boldsymbol{z}_j)}(x_{i,j}),\mu_{CS(p-1)}(x_{i,j})\} = \mu_{CS(p)}(x_{i,j})\end{aligned} \tag{A.15}$$

Similarly, for$x_{i,j}\in DS^{+}(\boldsymbol{z}_j)$ we have

$$\mu_{AC(p,\boldsymbol{z}_j)}(x_{i,j}) = \max\{0,\mu_{G(\boldsymbol{z}_j)}(x_{i,j})-\mu_{CS(p-1)}(x_{i,j})\} = \mu_{G(\boldsymbol{z}_j)}(x_{i,j})-\mu_{CS(p-1)}(x_{i,j}) \tag{A.16}$$

In view of (A.16), (A.13) becomes

$$\begin{aligned}\mu_{CS(p-1)\oplus AC(p,\boldsymbol{z}_j)}(x_{i,j}) &= \min\{1,\mu_{CS(p-1)}(x_{i,j})+\mu_{G(\boldsymbol{z}_j)}(x_{i,j})-\mu_{CS(p-1)}(x_{i,j})\} = \mu_{G(\boldsymbol{z}_j)}(x_{i,j})\\ &= \max\{\mu_{G(\boldsymbol{z}_j)}(x_{i,j}),\mu_{CS(p-1)}(x_{i,j})\} = \mu_{CS(p)}(x_{i,j})\end{aligned} \tag{A.17}$$

Combining (A.15) and (A.17), for every $x_{i,j}\in D$ we have

$$\mu_{CS(p-1)\oplus AC(p,\boldsymbol{z}_j)}(x_{i,j}) = \mu_{CS(p)}(x_{i,j}) \tag{A.18}$$

References

[1] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1997) 245–271.

[2] M. Ben-Bassat, Pattern recognition and reduction of dimensionality, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Handbook of Statistics-II, North Holland, 1982, pp. 773–791.

[3] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997).

[4] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (3) (1997) 131–156.

[5] Y. Kim, W. Street, F. Menczer, Feature selection for unsupervised learning via evolutionary search, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 365–369.

[6] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.

[7] J. Bi, K.P. Bennett, M. Embrechts, C.M. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, J. Mach. Learn. Res. (2003) 1229–1243.

[8] H. Liu, H. Motoda, in: Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publisher, Dordrecht, 1998.

[9] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. (2003) 1157–1182.

[10] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Netw. (1994) 537–550.

[11] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. (2005) 1226–1238.

[12] C.H. Ooi, M. Chetty, S.W. Teng, Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets, Data Min. Knowl. Discovery 114 (2007) 329–366.

[13] Y. Li, Z.F. Wu, Fuzzy feature selection based on min–max learning rule and extension matrix, Pattern Recognition 41 (2008) 217–226.

[14] K.Z. Mao, Orthogonal forward selection and backward elimination algorithms for feature subset selection, IEEE Trans. Syst. Man Cybern. B (2004) 629–634.

[15] X. Fu, L. Wang, Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance, IEEE Trans. Syst. Man Cybern. B (2003) 399–409.

[16] I. Kononenko, Estimating attributes: analysis and extensions of relief, in: Machine Learning: ECML-94, Lecture Notes in Computer Science, vol. 784, Springer, Berlin, Heidelberg, 1994, pp. 171–182.

[17] D Zhang, S Chen, Z.-H Zhou, Constraint score: a new filter method for feature selection with pairwise constraints, Pattern Recognition 41 (5) (2008) 1440–1451.

[18] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 301–312.

[19] S. Das, Filters, wrappers and a boosting-based hybrid for feature selection, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 74–81.

[20] A.Y. Ng, On feature selection: learning with exponentially many irrelevant features as training examples, in: Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 404–412.

[21] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (9) (1977) 917–922.

[22] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 153–158.

[23] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Lett. 15 (1994) 1119–1125.

[24] S. D. Stearns, On selecting features for pattern classifiers, in: Proceedings of the Third International Conference on Pattern Recognition, Coronado, CA, 1976, pp. 71–75.

[25] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discovery 2 (2) (1998) 121–167.

[26] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Netw. 12 (2) (2001) 181–201.

[27] Chih-Wei Hsu, Chih-Jen Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.

[28] D Dubois, H. Prade, in: Fuzzy Sets and Systems: Theory and Applications, Academic Press, 1980.

[29] A.H. Gunatilaka, B.A. Baertlein, Feature-level and decision-level fusion of non coincidently sampled sensors for land mine detection, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 577–589.

[30] Ludmilla Kuncheva, in: Combining Pattern Classifiers: Methods and Algorithms, John Wiley and Sons Inc., New Jersey, 2004.

[31] S.-B. Cho, J.H. Kim, Combining multiple neural networks by fuzzy integral for robust classification, IEEE Trans. Syst. Man Cybern. 25 (2) (1995) 380–385.

[32] L.I. Kuncheva, J.C. Bezdek, R. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognition 34 (2001) 299–314.

[33] Y. Freund, R.E. Shapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.

[34] F. Hoffmann, Combining boosting and evolutionary algorithms for learning of fuzzy classification rules, Fuzzy Sets Syst. 14 (2004) 47–58.

[35] UCI machine learning repository: 〈http://www.ics.uci.edu/~mlearn/MLRepository.html〉.

[36] J.C. Bezdek, in: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.

[37] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, et al., Molecular classification of cancer: class

discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[38] M. Shipp, K. Ross, P. Tamayo, A. Weng, J. Kutok, R. Aguiar, M. Gaasenbeeck, M. Angelo, M. Reich, G. Pinkus, et al., Diffuse large B-cell lymphoma outcome prediction by geneexpression profiling and supervised machine learning, Nat. Med. 8 (2002) 68–74.
[39] J. Khan, J. Wei, M. Ringner, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P. Meltzer, Classification and diagnostic prediction of cancers using expression profiling and artificial neural networks, Nat. Med. 7 (2001) 673–679.
[40] D. Li, W. Pedrycz, N.J. Pizzi, Fuzzy wavelet packet based feature extraction method and its application to biomedical signal classification, IEEE Trans. Biomed. Eng. vol. 52 (2005) 1132–1139.
[41] J. Platt, Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in: A.J. Smola, P. Bartlett, B. Schoelkopf, D. Schurmans (Eds.), Advances in Large Margin Classifiers, pp. 61–74, 2000.
[42] A. Madevska-Bogdanova, D. Nikolik, L. Curfs, Probabilistic SVM outputs for pattern recognition using analytical geometry, Neurocomputing 62 (1–4) (2004) 293–303.
[43] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and naïve Bayesian classifiers, in: Proceedings of the 18th International Conference on Machine Learning, ICML'01, 2001, pp. 609–616.
[44] J. Drish, in: Obtaining calibrated estimates from Support Vector Machines, Technical Report, University of California, San Diego, 2001.
[45] D. Bouchaffra, V. Govindaraju, S. Srihari, A methodology for mapping scores to probabilities, IEEE Trans. Pattern Anal. Mach. Intell. 21 (9) (1999) 923–927.
[46] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 694–699.
[47] A. Niculescu-Mizil, R. Caruana, Predicting good probabilities with supervised learning, in: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 625–632.
[48] N. Kwak, C-H Choi, Improved mutual information feature selector for neural networks in supervised learning, in: Proceedings of the Joint Conference on Neural Networks, Washington, DC, July 1999.

Serafeim P. Moustakidis was born in Thessaloniki, Greece, in 1981. He received a degree in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2004. He is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, Division of Electronics and Computer Engineering, Aristotle University of Thessaloniki. His research interests include fuzzy logic systems, support vector machines, wavelet analysis, control with neural networks, and genetic algorithms.

John B. Theocharis received a degree in electrical engineering and the Ph.D. degree from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1980 and 1985, respectively. He is currently a Professor in the Department of Electronic and Computer Engineering, Division of Electronics and Computer Engineering, Aristotle University of Thessaloniki. His research activities include fuzzy systems, neural networks, neuro-fuzzy modeling, time-series prediction, recurrent fuzzy neural networks, evolutionary algorithms, and fuzzy pattern recognition.