A General Wrapper Approach to Selection of Class-Dependent Features

Lipo Wang, Nina Zhou, and Feng Chu

***Abstract*—In this paper, we argue that for a C-class classification problem, C 2-class classifiers, each of which discriminating one class from the other classes and having a characteristic input feature subset, should in general outperform, or at least match the performance of, a C-class classifier with one single input feature subset. For each class, we select a desirable feature subset, which leads to the lowest classification error rate for this class using a classifier for a given feature subset search algorithm. To fairly compare all models, we propose a weight method for the class-dependent classifier, i.e., assigning a weight to each model's output before the comparison is carried out. The method's performance is evaluated on two artificial data sets and several real-world benchmark data sets, with the support vector machine (SVM) as the classifier, and with the RELIEF, class separability, and minimal-redundancy–maximal-relevancy (mRMR) as attribute importance measures. Our results indicate that the *class-dependent* feature subsets found by our approach can effectively remove irrelevant or redundant features, while maintaining or improving (sometimes substantially) the classification accuracy, in comparison with other feature selection methods.**

***Index Terms*—Class-dependent feature extraction, class-dependent feature selection, feature importance ranking, minimal-redundancy–maximal-relevancy (mRMR), one-against-all, one-against-one, support vector machine (SVM).**

I. Introduction

TODAY data sets that we process are becoming increasingly larger, not only in terms of the number of patterns (instances), but also the dimension of features (attributes), which may degrade the efficiency of most learning algorithms, especially when there exist irrelevant or redundant features. Langley *et al.* [32], [33], [47] pointed out that the predictive accuracy of the learning algorithms are reduced in the presence of irrelevant features. Koller *et al.* [29] proved that the distribution of truly relevant features for the main task are blurred by irrelevant or redundant features [45]. Fu and Wang [17] showed that deleting those irrelevant features cannot only improve the classification accuracy, but also reduce the structural complexity of the radial basis function (RBF) neural network and facilitate rule extraction. All these reasons urge us to carry out data dimensionality reduction (DDR) [21], [22], [45], [55].

There are numerous techniques for DDR. Depending on whether the original features are transformed to new features, one may categorize these techniques into feature extraction or feature selection techniques, respectively. Depending on whether a classifier is used to evaluate the performance of the new feature set, these DDR techniques can be categorized into wrapper or filter methods, respectively. Feature extraction methods, e.g., principal component analysis (PCA) [10] and linear discriminant analysis (LDA) [38], [39], [46], transform the original set of features into a new set of features. Because the new features are different from the original features, it may be difficult to interpret the meaning of the new features. Feature selection [55] selects a desirable subset of the original features while maintaining or increasing acceptable classification accuracy. Feature selection eliminates unimportant features, e.g., redundant and irrelevant features, and obtains the best subset of features that discriminates well among classes. Thus how to decide which features are important so as to form the desirable feature subset is the main objective of feature selection.

In feature selection, one usually chooses the same feature subset for all classes in a given classification problem, which is called *class-independent* feature selection [17], [27], [46], [49], [50]. In contrast, one may also allow for a different feature subset for each class, which is called *class-dependent* feature selection [41], [42], since different features may have different capabilities in discriminating different classes. *Class-independent* feature selection commonly practiced can, therefore, be considered as a special case of the more general *class-dependent* feature selection.

Let us consider the following scenario. Suppose we have data on patient information, such as age, body weight, height, blood pressure, cholesterol, race, etc., and we need to use these data for diagnosing of three types of diseases, say A, B, and E. Suppose there are underlying scientific reasons (not yet unknown to the world) that 1) a patient has disease A if and only if the body weight of the patient is above a certain limit; 2) a patient has disease B if and only if the blood pressure of the patient is below a certain limit; 3) a patient has disease E if and only if the cholesterol is within a certain range; and 4) all the other patient attributes are irrelevant to this diagnostic task. For this four-class classification problem, with the four classes being disease A, disease B, disease E, and healthy (for the sake of simplicity, we assume that it is impossible to have more than one of these diseases simultaneously), one may train either one four-class classifier, or four two-class (binary) classifiers, each of which discriminates one class from the other three classes. Suppose an effective feature selection technique is used. The single four-class classifier approach should yield three input features, i.e., body weight, blood pressure, and cholesterol, whereas the approach with four two-class classifiers should yield body weight for the disease-A classifier, blood pressure for the disease-B classifier, cholesterol for the disease-E classifier, and body weight, blood pressure, and cholesterol for the healthy classifier. All the other input attributes are noise and should have been removed by the

Manuscript received July 29, 2007; revised November 27, 2007; accepted December 16, 2007. First published May 30, 2008; last published July 7, 2008 (projected).

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: elpwang@ntu.edu.sg).

Digital Object Identifier 10.1109/TNN.2008.2000395

feature selection algorithm. We say that the four-class classifier has *class-independent* input features, whereas the approach with four two-class classifiers has *class-dependent* input features. In general, we have the following reasons to expect the approach with four two-class classifiers (using *class-dependent* feature subsets) to outperform the single four-class classifier (using *class-independent* feature subsets): If an input data represents a patient with disease A, the blood pressure and cholesterol part of the data will act as noise to the multiclass classifier, whereas if an input data represents a patient with disease B, the body weight and cholesterol part of the data will act as noise to the multiclass classifier. In contrast, there will be no noise for the approach with four two-class classifiers (we will later demonstrate this line of arguments with simulations).

There has been extensive research effort on *class-independent* feature selection (see, e.g., references listed in [17]). For example, Kira *et al.* proposed RELIEF [27] to select the best set of features for all classes. They use the "nearest hit" and "nearest miss" (for more details see Section III) to calculate a weight for each feature and rank the features according to the weights which reflect features' relevance. The original RELIEF was limited to only two-class problems. Later in [30], Kononenko extended RELIEF to RELIEF-F (or RELIEFF), which is able to deal with multiclass problems and incomplete data. Both Kira [27] and Kononenko [30] adopted the filter approach to selecting class-independent features and the goodness of the selected feature subsets depended on the choice of a relevance threshold. In [49], Roberto proposed a filter approach to selection of attributes by projection (SOAP), which first evaluated the feature importance by counting the label changes of each attribute produced when crossing the projections of each example in each dimension, and then selected a number of features according to a threshold. This feature selection method is very fast but leads to rather low classification accuracy [23], [49]. Siedlecki *et al.* [50] and Raymer *et al.* [46] proposed a wrapper approach to selecting class-independent feature subsets using the genetic algorithm (GA), where GA is used to find a binary vector (feature mask) and each bit represents the presence (1) or absence (0) of a feature. The feature subset was obtained according to the nearest-neighbor classifier. Fu and Wang [17] proposed a method of attribute importance ranking using a separability-correlation measure (SCM), which is the combination of a class separability measure (CSM) and an attribute-class correlation measure. According to the SCM, attribute subsets are selected based on the classification accuracy of an RBF classifier.

Class-independent feature selection neglects the possibility that different groups of features may have different power in distinguishing different classes. To look for any advantages that this possibility may bring about, one may attempt to use *class-dependent* feature selection techniques, which select a different feature subset for each class. Baggentstoss [2], [3] proposed to select class-specific features on the basis of the probability density function (pdf) projection theorem. Baggenstoss [2], [3] also provided theoretical proof for this method and demonstrated applications on signal processing problems. Oh *et al.* proposed [41], [42] a *filter approach* to selecting class-dependent features for handwriting digits. They used the estimated class distributions to calculate each feature's class separation for the ten digits (classes). Then, in terms of class separation, an ordered list of features was provided for each class and according to the ranking list each class obtained a feature vector with a predefined dimension 256. Although all the ten feature vectors have the same dimension, they have different feature compositions. In [18], Fu and Wang used GA to select a feature subset for each class based on an RBF classifier. This is a wrapper approach to class-dependent feature selection, however, this approach made explicit use of the clustering property of the RBF neural network and therefore may not work for other types of classifiers, for example, the support vector machine (SVM) and the multilayer perceptron (MLP) neural network. Besides, class-dependent feature extraction methods were proposed. For example, Liu *et al.* [36] proposed to extract class-specific features through principle component analysis (PCA) from class-specific subspaces.

All the aforementioned feature selection and feature extraction methods can also be classified into filter approaches [27], [30], [36], [41], [42], [49] and wrapper approaches [17], [18], [46]. Filter approaches select feature subsets independent of a classifier, whereas wrapper approaches select features using a classifier. Due to this difference, wrapper approaches tend to have better classification performance compared to filter approaches.

We structure the remainder of this paper as follows. In Section II, we propose our general wrapper approach to selecting *class-dependent* features that will work with any arbitrary classifier. In Section III, we present experimental results on two artificial data sets, four data sets from the University of California at Irvine (UCI) database [5], and two data sets from the Library for Support Vector Machines (LIBSVM) data [9], and compare the results of *class-independent* and *class-dependent* feature selection, *class-independent* and *class-dependent* feature extraction, all using the strategy of "one-against-all," together with class-dependent feature selection using the strategy of "one-against-one." In Section IV, we provide a summary and discussion.

II. General Methodology

Our general wrapper approach to selection of class-dependent features consists of the following three steps (Fig. 1). In the first step, we convert a C-class classification problem to C two-class classification problems, i.e., problem 1, problem 2, ..., and problem C. The goal of problem c is to correctly separate the original class c from the other patterns, where $c = 1, 2, \ldots, C$. For problem c, we divide all training patterns into two classes: class A is the original class c, and class B consists of all the other training patterns.

The second step is to search for a desirable feature subset for each binary classification problem. In this paper, we will adopt a forward search as follows. We rank the attribute importance using a ranking measure, such as RELIEF weight measure [27], CSM [15], [17], information-theoretic measure [34], and the minimal-redundancy–maximal-relevancy (mRMR) measure [43] based on mutual information (MI). In this paper, we adopt the RELIEF weight measure, CSM, and the mRMR measure [43] to evaluate the importance of the features for each class (in each of the C subproblems). We will briefly review these three feature ranking measures in the next section. We regard

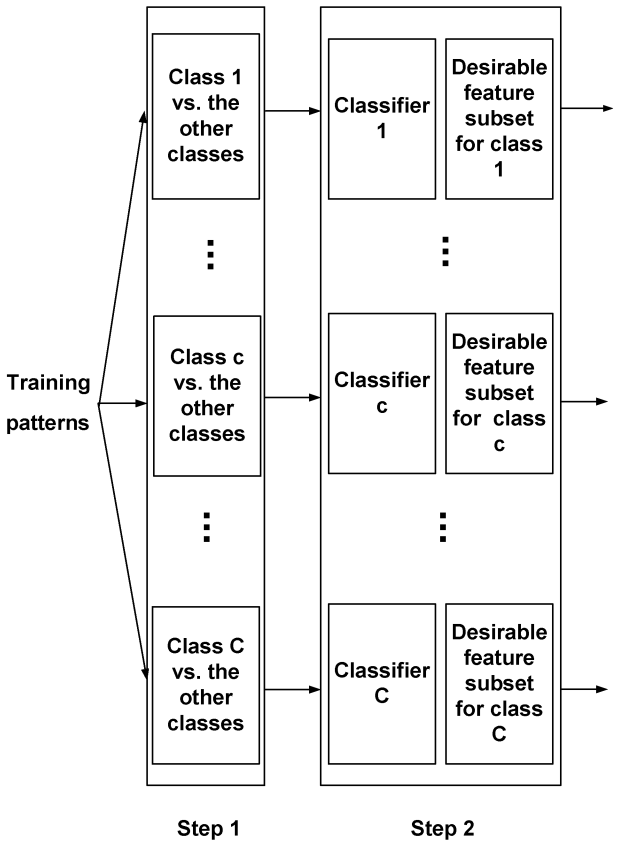

Fig. 1. Schematic diagram for our general wrapper approach to class-dependent feature selection.

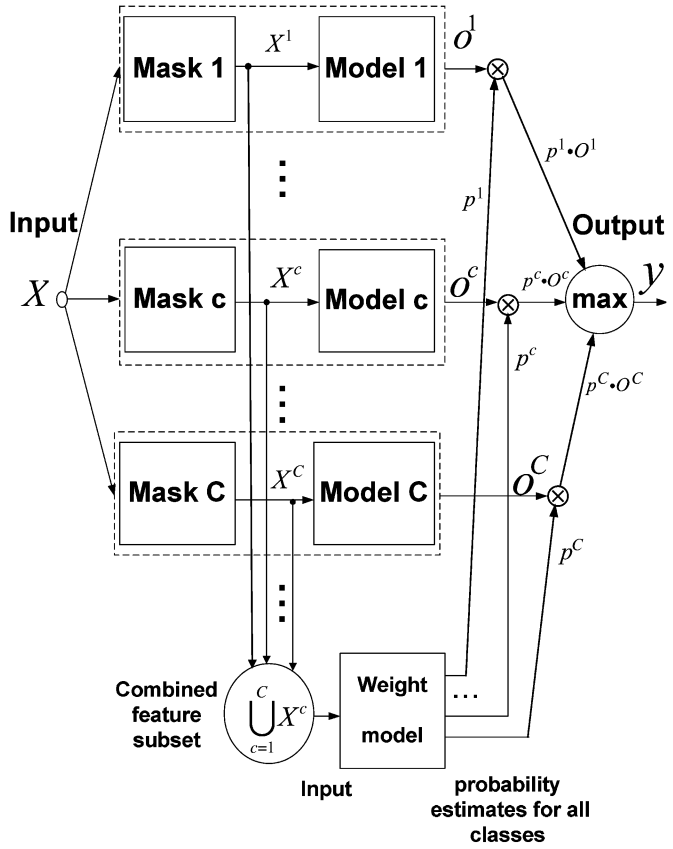

Fig. 2. Architecture of the general classifier with class-dependent feature subsets (after training and during testing).

the attribute importance ranking thus obtained *class-dependent attribute importance ranking*.

After obtaining the attribute importance ranking of each class, we then choose a desirable feature subset for each class through a classifier, e.g., the support vector machine (SVM), for a given feature subset search algorithm. For each class, we use forward selection search (or bottom-up search) to form different attribute subsets [17], that is, we start with the most important feature as the first feature subset, and then each time add one attribute into the previous subset in the order of importance ranking to form a new feature subset, until a stopping condition is satisfied, e.g., the validation accuracy starts to decrease or all the attributes in this class have been added. For high-dimensional data sets, it will be computationally expensive to incrementally form all feature subsets and evaluate them through a classifier. Therefore, we will first select certain number of features, e.g., the top 70 from 649 ranked features for the handwritten digits recognition (HDR)-multifeature (or HDR) data set from the LIBSVM data [9], and then incrementally form different feature subsets. Inputting each feature subset into the classifier, we can obtain different classification accuracy for different feature subsets. Then we choose the attribute combination with the highest classification accuracy or lowest error rate as the desirable feature subset for a given feature subset search algorithm.

We use a feature mask to represent a feature subset, i.e., a "0" or "1" in a feature mask indicates the absence or presence of a particular feature, respectively. For example, if originally there are five features, i.e., $\{x_1, x_2, x_3, x_4, x_5\}$, and the desirable feature subset turns out to be $\{x_1, x_2, x_3\}$ with the fourth and fifth features deleted, the feature mask will be $\{1, 1, 1, 0, 0\}$.

In the testing stage after training, we classify the input test pattern according to the classifier (Fig. 2) with the maximum response. The classifier in Fig. 2 includes several models. When using those class-dependent feature subsets to train and test models, each model has an output, which provides the probability that each sample belongs to the current class in terms of the current class' feature subset [36]. If we use SVM in all models, we have the probability estimate [9], [57] of each current class as the output of each model. According to Chang *et al.* [9] and Wu *et al.* [57], in SVM models, each class has a probability estimate which indicates the probability the testing sample belongs to each class. A common way to generate an overall output from individual outputs of all models is to choose the maximum individual as the overall output of the class-dependent classifier, e.g., as Oh *et al.* [41] and Liu *et al.* [36] mentioned. However, Baggenstoss pointed out that it is not fair to directly compare outputs produced from different feature spaces [2], [3]. Baggenstoss [2] proposed the pdf projection theorem and utilized it to project those class-specific features' pdfs back into the original space where a fair comparison is possible. In our paper, we propose a heuristic method, i.e., a weight measure, to deal with the problem of unfair comparisons on outputs from different feature spaces.

Suppose the output of the cth model is denoted as O^c $(c = \{1, 2, \ldots, C\})$, which is the probability that one sample

belongs to the cth class when using the cth feature subset obtained by the cth model to classify the sample between two classes, i.e., class A (the cth class) and class B (all other classes except the cth class). For C models, we have C outputs, i.e., $O^1, O^2, \ldots, O^c, \ldots, O^C$, each representing the probability that the sample belongs to the current class, respectively. Instead of comparing the C outputs obtained from different feature spaces, we assign a weight to each output and compare the weighted outputs, e.g., $p^c \cdot O^c$ in Fig. 2. These weights, i.e., $p^1, \ldots, p^c, \ldots, p^C$ in Fig. 2, are obtained on the basis of a common feature space. That is, we first combine all class-dependent feature subsets together to form one union set, i.e., $\cup_{c=1}^{C} X^c$. Then we classify the sample using the union set as the feature subset with C classes as target classes, together with a classifier, such as an SVM. The classifier produces outputs $p^1, \ldots, p^c, \ldots, p^C$, each representing the probability that the sample belong to each of C class, respectively. We assign these probabilities obtained from the common feature space to the corresponding $O^1, \ldots, O^c, \ldots, O^C$ for a fair comparison. Since class-dependent feature subsets vary with different training samples in the training process, those weights also vary with different leaning processes. Note that the weight model in Fig. 2 has C outputs, i.e., probability estimates for all classes.

III. Experimental Results

We will use two criteria to compare the performance of our approach to *class-dependent* feature selection with other feature selection and feature extraction methods, as well as those without feature selection or feature extraction. The two criteria are as follows.

- *Dimensionality Reduction*. By contrasting the number of original features, we can see how many features are selected. In our approach, we determine the number of selected features for each class in terms of the average number with the standard deviation in all simulations.
- *Classification Accuracy*. We expect that our approach should maintain or even improve on the accuracy, after removing redundant or irrelevant features that may degrade the classification accuracy.

A. Review of Three Feature Importance Ranking Measures Used in This Paper

1) *RELIEF Weight Measure:* RELIEF proposed by Kira and Rendell [27] is a filter approach to feature selection and has been shown to be very efficient in evaluating features importance. The evaluation measure to features is based on how well those features distinguish among instances that are close to each other. Here two notations are used to describe the two nearest neighbors of the given instance: *nearest hit* (nearest neighbor of the given instance from the same class) and *nearest miss* (nearest neighbor of the given instance from different classes). The original *RELIEF* algorithm is depicted as follows.

Step 1) Introduce a weight vector and initialize it to zero: $\{w_1, \ldots, w_i, \ldots, w_D\} = 0$. D is the number of features.

Step 2) Randomly select an instance $\mathbf{X}$ from training instances S and find its *nearest hit* $\mathbf{X}_h$ and *nearest miss* $\mathbf{X}_m$.

Step 3) Calculate and update the weight w_i of the ith feature $x_i : w_i = w_i + (x_i - x_{mi})^2 - (x_i - x_{hi})^2, i = 1, 2, \ldots, D$. x_{mi} is the ith element of $\mathbf{X}_m$ and x_{hi} is the ith element of $\mathbf{X}_h$. Each feature's weight is updated by the differences (Euclidean distance) between the samples and the misses (miss differences) and between the samples and the hits (hit differences). The weight is increased if the miss differences are greater than the hit differences, which means that the feature is relevant.

Step 4) Repeat Steps 2) and 3) over all training instances.

In our approach to class-dependent feature selection, multiclass problems are already converted into multiple two-class problems, and therefore we can use the original RELIEF algorithm [27] to rank the features. However, for class-independent feature selection method, it is a multiclass problem so that we will use the extended version RELIEFF [30] to rank the features importance. Each sample will have one hit and more than one miss, and the differences between the misses and the sample are weight-averaged. The weight adopted here is the *a priori* probability of the miss class.

After obtaining the weight of each feature, we rank the importance of the features according to the rationale: the larger the weights, the more important the features.

2) *Class Separability Measure:* The CSM [15], [17] evaluates how well two classes are separated by a feature vector. The greater the distance is between different classes, the easier the classification task. Therefore, the feature subset that can maximize the distances between different classes may also maximize classification accuracy and is therefore considered more important. The jth sample is represented as $\{\mathbf{X}_j, t_j\}$, where $\mathbf{X}_j = \{x_{j1}, x_{j2}, \ldots, x_{jD}\}$ is the input data and $t_j = \{1, 2, \ldots, C\}$ is the class label of $\mathbf{X}_j$. Class separability consists of two elements, i.e., the distance between patterns within each class S_w [17]

$$S_w = \sum_{c=1}^{C} P_c \sum_{j=1}^{n_c} \left[(\mathbf{X}_{cj} - \overline{m}_c)(\mathbf{X}_{cj} - \overline{m}_c)^T \right]^{1/2} \quad (1)$$

and the distance between patterns among different classes S_b [17]

$$S_b = \sum_{c=1}^{C} P_c \left[(\overline{m}_c - \overline{m})(\overline{m}_c - \overline{m})^T \right]^{1/2}. \quad (2)$$

Here P_c is the probability of the cth class and n_c is the number of samples in the cth class. $\mathbf{X}_{cj}$ is the jth sample in the cth class, $\overline{m}_c$ is the mean vector of the cth class, and $\overline{m}$ is the mean vector of all samples in the data set.

The ratio S_w/S_b can be used to measure the separability of the classes [17]: the smaller the value, i.e., the smaller the distances within each class and the greater the distance among different classes, the better the separability. The importance of a feature may be evaluated by ratio S_w/S_b calculated *after the feature is removed from the data set*, i.e., S'_w/S'_b. The greater S'_w/S'_b is, the more important the

removed attribute is. For example, if removing attribute 1 from the data set leads to greater S'_w/S'_b, compared with removing attribute 2, we may consider attribute 1 more important compared to attribute 2 for classifying the data set, and vice versa. Hence, we may evaluate the importance level of the attributes according to ratio S_w/S_b [17] with an attribute deleted each time in turn.

3) *The Minimal-Redundancy–Maximal-Relevancy Measure:* The mRMR feature selection method was proposed by Peng *et al.* [43]. It is based on the MI theory to select features in terms of their relevancy and redundancy.
Let $F_D = \{x_i | i = 1, 2, \ldots, D\}$ denote a feature set. According to the definition of MI in [12], the MI value of two features x_i and x_j is denoted as $I(x_i; x_j)$ $(i, j = 1, 2, \ldots, D)$, which describes statistical dependence between the two features. In the same manner, the MI value $I(x_i; c)$ (c is one of C classes) is used to denote the statistical dependency of the feature x_i to the class c. Peng *et al.* [43] used the incremental search strategy to sequentially select features with the maximal relevancy and minimal redundancy [see (3)]. The sequence in which features are included corresponds to the ranking order of features. Suppose F_{m-1} denote the feature subset consisting of $m-1$ top ranked features, the mth ranked feature x_m should be selected according to the following:

$$\max_{x_m \in (F_D - Fm-1)} \left[I(x_m; c) - 1/m \sum_{x_j \in F_{m-1}} I(x_m; x_j) \right]. \quad (3)$$

Although RELIEF cannot detect redundancy in features, RELIEF was often adopted due to its efficiency in computation [4]. The CSM is used to detect feature's relevancy in classification problems by many authors, e.g., Fu *et al.* [17], even as a linear ranking measure. Compared with these two feature importance ranking measures, the mRMR measure may be more effective since it is a nonlinear method and it evaluates features in terms of both relevancy and redundancy.

B. Feature Extraction Method—Kernel Fisher Discriminant Analysis

We also adopt a class-dependent feature extraction method for comparison. Among those common feature extraction methods, e.g., PCA [10], [36], Fisher discriminant analysis (FDA) and its variant kernel FDA [37], [39], [40], we choose kernel FDA (KFDA) in our experiment, since KFDA is a class-specific transformation and is hence expected to be superior in classification problems.

The KFDA was proposed by Mika *et al.* [39]. It first maps the original data into a feature space, and then conducts the FDA algorithm in the feature space to extract nonlinear discriminative features from the original data. According to [39], the KFDA algorithm is described as maximizing the following objective:

$$J(w) = \frac{w^T M_B^\phi w}{w^T M_W^\phi w}. \quad (4)$$

Here M_B^ϕ is the between-class scatter matrix in the feature space

$$M_B^\phi = \sum_{c=1}^{C} P_c \left[\left(\overline{m}_c^\phi - \overline{m}^\phi \right)^T \left(\overline{m}_c^\phi - \overline{m}^\phi \right) \right] \quad (5)$$

and M_W^ϕ is the within-class scatter matrix in the feature space

$$M_W^\phi = \sum_{c=1}^{C} P_c \sum_{j=1}^{n_c} \left[\left(\mathbf{X}_{cj}^\phi - \overline{m}_c^\phi \right)^T \left(\mathbf{X}_{cj}^\phi - \overline{m}_c^\phi \right) \right]. \quad (6)$$

ϕ is the transformation function, which realizes the kernel trick. In this case, $X_{cj}^\phi = \phi(X_{cj})$, $\overline{m}_c^\phi = 1/n_c \sum_{j=1}^{n_c} \phi(X_{cj})$, and $\overline{m}^\phi = 1/N \sum_{j=1}^{N} \phi(X_j)$. X_j and X_{cj} are the same as previous descriptions in the CSM measure. N is the total number of samples. Common kernel functions include Gaussian RBF, polynomials, and so on. In this paper, we adopt the Gaussian RBF kernel, which is denoted as $k(x, y) = \exp(-(\|x-y\|^2/\delta))$ and δ is a positive constant. In our experiment, we have $\delta = 2^{-7}$.

C. One-Against-One Strategy for Class-Dependent Feature Selection

In our experiment, the "one-against-all" strategy [25] is used in previous class-dependent feature selection and feature extraction methods. The "one-against-one" strategy [16], [25], [44] can also be used to realize the class-dependent feature selection. The basic theory of "one-against-one" is to construct all possible two-class classifiers, i.e., $C(C-1)/2$ classifiers for C classes. Each classifier is trained on samples only from two classes. When testing, we count the vote of each class and adopt the Max Wins algorithm [16] to determine the final output. Due to space and time limitation, in our experiment, we combine this "one-against-one" strategy only with the mRMR ranking measure to select class-dependent features.

D. Data Sets

To demonstrate the scenario given in the Introduction, we generate two artificial data sets. Data set A has 750 samples with five classes and ten input features $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$. x_i is useful in discriminating only class i with $i = 1, 2, \ldots, 5$. The feature x_6 is relevant to all the classes, whereas x_7, x_8, x_9, and x_{10} are irrelevant to all the five classes. For example, x_1 is randomly generated with a mean of 1 that is distributed with a variance of 0.001 for the class 1 and random values within $[0.01, 30]$ with a great variance for other classes except the class 1. x_2 is randomly generated with a mean of three that is distributed with a variance of 0.001 for the class 2 and random values within $[0.01, 30]$ for other classes except the class 2. Similarly, x_i with $i = 3, 4, 5$ is randomly generated. x_6 is randomly generated with five different mean values for five classes and each mean is distributed with a variance of 0.0001. x_i with $i = 7, 8, 9, 10$ is randomly generated within $[0.01, 30]$ with a great variance. Data set B has 1500 samples with ten classes and 20 features. Feature i is useful in discriminating only class i with $i = 1, 2, \ldots, 10$. The feature 11 is relevant to all the classes, whereas all the left features from 12 to 20 are irrelevant to all the ten classes. All features are randomly generated in the

TABLE I
BASIC INFORMATION OF THE DATA SETS USED IN THIS PAPER

Data set	Number of samples	Total number of classes	Total number of attributes
Artificial data A	750	5	10
Artificial data B	1500	10	20
Waveform40	1000	3	40
Segment	2310	7	19
Vehicle	846	4	18
HDR-multifeature	2000	10	649
DNA	3186	3	180
Scene	2407	6	294

same manner as the artificial data A. We apply our class-dependent feature selection method with the CSM and RELIEFF ranking measures, respectively, both using the "one-against-all" strategy on the two artificial data sets and three low-dimensional data sets from the UCI Machine Learning Repository [5], i.e., Waveform, Segment, and Vehicle data sets. We apply class-dependent feature selection methods with the CSM and mRMR ranking measure, respectively, class-dependent feature extraction methods, all using the "one-against-all" strategy, and the class-dependent feature selection with the mRMR ranking measure using the "one-against-one" strategy (Table VII) on three high-dimensional data sets, i.e., the HDR data set from the UCI Machine Learning Repository [5], the DNA and the Scene data from the LIBSVM data [9]. The data sets and their characteristics are summarized in Table I. Since the Waveform data set has two groups of data and the group used in our experiment includes 40 features, we use Waveform40 to denote it.

E. Results and Discussions

We train and test the two artificial data sets and three low-dimensional data sets, i.e., the Waveform40, Segment, and Vehicle data set, using tenfold cross validation, that is, we randomly divide each data set into ten subsets of equal size; each time we use nine subsets for training and the other one for testing. For the three high-dimensional data sets, we use a different training and testing method, i.e., randomly separating data into training and testing part. For example, for the HDR data set, we randomly select 1000 samples for training and 1000 for testing. For the DNA data set, we randomly select 2000 samples for training and 1186 for testing. For the Scene data set, we randomly select 1211 for training and 1196 for testing. The whole training and testing processes are repeated ten times.

In Table II, we present the numbers of times that each feature of the artificial data A and B is selected in the ten simulations, the average number of features selected, and the standard deviation for class-independent and class-dependent feature selection methods with CSM ranking measure and the "one-against-all" strategy. We can see that, for both the class-independent and class-dependent feature selection methods, irrelevant features 7–10 are always eliminated for artificial data A, and irrelevant features 12–20 are always eliminated for artificial data B. These results show that the two feature selection methods are very effective in deleting noises (irrelevant features). On the other hand, there are some differences between the results obtained from these two methods. For *class-independent* feature selection method, features 1, 3, 5, and 6 are selected in ten simulations and features 2 and 4 are selected in seven simulations for the artificial data A, while for *class-dependent* feature selection method, each class has its own specific feature subset. For example, classes 1, 2, and 5 have stable feature subsets $\{1,6\}, \{2,6\}$, and $\{5,6\}$, respectively, in ten simulations. The feature subsets for classes 3 and 4 have variations in ten simulations. The class-independent and class-dependent feature selection methods nearly produce the same difference on artificial data B as on artificial data A. When comparing the average number of features selected using the class-dependent feature selection method with that of the class-independent feature selection method, the *class-dependent* feature selection method select much fewer features than the *class-independent* feature selection method. For example, artificial data A has on average 5.5 features selected for class-independent feature selection method, while for class-dependent feature selection method, it has on average 1.8 features selected (see the last row of the Table II). Artificial data B has on average 1.6 features selected for class-dependent feature selection method and 5.4 features selected for class-independent method. After considering the standard deviation, the conclusion is still the same.

For the three low-dimensional data sets from the UCI machine learning database, we respectively utilize the RELIEF and CSM to rank features for the class-independent and class-dependent methods using the "one-against-all" strategy. From the results in Table III, we can see that the Vehicle data set has few irrelevant or redundant features, whereas the Waveform40 and Segment data sets have many irrelevant or redundant features deleted by the class-independent and class-dependent methods. More importantly, our class-dependent feature selection leads to smaller feature subsets in comparison with class-independent feature selection, especially for the CSM measure. For example, the Waveform40 data set has on average 24 features selected for the class-independent method with the CSM measure, while for the class-dependent method, the average numbers of features selected for the three classes are 19. The Segment data set has on average 15 features selected by the class-independent method with the RELIEF measure, 19 features selected by the CSM measure, while on average ten features selected using the class-dependent method with the RELIEFF measure and 11 features using the CSM measure (see the last row of Table III). Also rather different numbers of features are selected for the seven classes by the class-dependent method.

For the three high-dimensional data sets, we utilize the mRMR and CSM ranking measures, respectively. The relevant results are presented in Table VI. For the HDR data set, the class-independent feature selection method with the mRMR measure selects on average 48 features from 649 features, and the class-dependent method with the mRMR measure selects only 21 features averaged from all class-dependent features. In comparison, the CSM ranking measure has on average 65 features selected by the class-independent and 33 features selected by the class-dependent method. For the Scene and DNA data sets, the class-dependent feature selection with both

TABLE II

TOTAL NUMBERS OF TIMES EACH FEATURE IS SELECTED IN TEN SIMULATIONS FOR ARTIFICIAL DATA A AND DATA B. FOR EXAMPLE, THE FIRST ROW OF DATA SHOWS THAT IN THE TEN SIMULATIONS, THE CLASS-INDEPENDENT FEATURE SELECTION METHOD SELECTED THE FIRST FEATURE TEN TIMES, THE SECOND FEATURE SEVEN TIMES, THE THIRD FEATURE TEN TIMES, THE FOURTH FEATURE SEVEN TIMES, . . . THE LAST THREE FEATURES (UNDERLINED) WERE ALMOST NEVER SELECTED. THE TABLE ALSO PRESENTS THE AVERAGE NUMBERS OF FEATURES SELECTED ± STANDARD DEVIATION (STD) IN TEN SIMULATIONS. THE CSM RANKING AND "ONE-AGAINST-ALL" ARE USED

Feature selection approach	Classes	Artificial Data A		Artificial Data B	
		Total numbers of times each of 10 features is selected in 10 simulations	Average number of features selected ±std	Total numbers of times each of 20 features is selected in 10 simulations	Average number of features selected ±std
Class-independent	all classes	10 7 10 7 10 10 1 0 0 0	5.5±1	1 8 4 1 4 7 4 5 6 4 10 0 0 0 0 0 0 0 0 0 0	5.4±2.2
Class-dependent	class 1	10 0 0 0 0 10 0 0 0 0	2	10 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0	1.3±0.5
	class 2	0 10 0 0 0 10 0 0 0 0	2	0 10 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0	2
	class 3	1 1 10 2 0 2 0 0 0 0	1.6±1.3	0 0 10 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	1.1±0.3
	class 4	0 0 0 10 1 1 0 0 0 0	1.2±0.6	0 0 0 10 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0	2
	class 5	0 0 0 0 10 10 0 0 0 0	2	0 0 0 0 10 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	1.2±0.6
	class 6	/	/	0 0 0 0 0 10 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	1
	class 7	/	/	0 0 0 0 0 0 10 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0	2
	class 8	/	/	0 0 0 0 0 0 0 10 0 0 9 0 0 0 0 0 0 0 0 0 0 0	1.9±0.3
	class 9	/	/	0 0 0 0 0 0 0 0 10 0 9 0 0 0 0 0 0 0 0 0 0 0	1.9±0.3
	class 10	/	/	0 0 0 0 0 0 0 0 0 10 10 0 0 0 0 0 0 0 0 0 0 0	2
	Average number±std		1.8±0.6	/	1.6±0.3

TABLE III

AVERAGE NUMBER OF FEATURES SELECTED IN TEN SIMULATIONS ± STD, BY CLASS-INDEPENDENT AND CLASS-DEPENDENT METHODS WITH THE CSM AND RELIEFF RANKING MEASURE AND "ONE-AGAINST-ALL" STRATEGY, FOR THE WAVEFORM40, SEGMENT, AND VEHICLE DATA SETS

Feature selection approach	Classes	CSM ranking			RELIEFF ranking		
		Waveform40	Segment	Vehicle	Waveform40	Segment	Vehicle
Class-independent	all classes	23.8±3	18.5±0.8	17.1±1	23.8±7.4	15.3±1.1	17.6±0.7
Class-dependent	class 1	19.3±3	8.6±0.2	17.5±0.8	23.9±4.8	8.8±0.4	16.2±1.9
	class 2	16.7±1	2.6±0.8	17.3±0.8	22.9±4.8	2.5±0.9	16.0±2.1
	class 3	20±4	15.8±0.6	16.8±1	23.1±5.9	14.3±2.3	17.1±1
	class 4	/	8.1±0.3	18±0	/	11.3±3.2	17.2±1
	class 5	/	17.1±0.3	/	/	12.3±1.6	/
	class 6	/	13.2±3.7	/	/	10.7±1.3	/
	class 7	/	3.4±1.3	/	/	6.9±3.8	/
	Average	18.7±2.9	11.2±2.3	17.4±0.8	23.3±5.2	10±2.5	16.6±1.6

ranking measures, i.e., the mRMR and CSM, selects fewer features than the class-independent feature selection method. When comparing the two ranking measures, we can see that the mRMR ranking measure shows advantages over the CSM ranking in terms of the average number of features selected.

Based on the above results, we still cannot say the class-dependent feature selection method outperforms the class-independent feature selection method. The key point we mostly care about is whether the classification accuracy will be increased more by the class-dependent feature selection method than by the class-independent method. We present the classification accuracy in Tables IV and VII, where the proposed general classifier is used for the class-dependent feature selection method and the normal SVM is used for the class-independent method. In Table IV, we can see that for the two artificial data sets both the *class-independent* and *class-dependent* feature selection methods have greatly increased the classification accuracy, and the *class-dependent* feature selection method has increased more than the *class-independent* feature selection method. The same thing happens for the Waveform40 data set. For the Segment and Vehicle data sets, although the increase on the accuracy is not so obvious as that for the two artifi-

TABLE IV
COMPARISONS OF CLASSIFICATION ACCURACY AMONG CLASS-INDEPENDENT AND CLASS-DEPENDENT FEATURE SELECTION METHODS WITH OUTPUT PROBABILITY NOT WEIGHTED AND WEIGHTED (FIG. 2), WITH THE CSM RANKING MEASURE AND "ONE-AGAINST-ALL" STRATEGY, FOR THE ARTIFICIAL DATA A AND B, WAVEFORM40, SEGMENT, AND VEHICLE DATA SETS. THE TABLE ALSO PRESENTS THE CLASSIFICATION RESULTS BY THE RELIEFF RANKING MEASURE WITHOUT USING THE WEIGHT APPROACH FOR THE WAVEFORM40, SEGMENT, AND VEHICLE DATA SETS

Feature selection approach (Wrappers)		Mean accuracy in 10 simulations ±std (minimum/maximum accuracy)				
		Artificial data A	Artificial data B	Waveform40	Segment	Vehicle
Without feature selection		93.87%±1.69% (90.67%/96%)	84.53%±2.10% (82%/87.33%)	84.50%±2.22% (80%/86%)	96.28%±1.16% (95.24%/98.70%)	79.43%±3.28% (75.29%/85.88%)
CSM + Without weighted	Class-independent	96.53%±2.28% (93.33%/100%)	91.53%±1.60% (89.33%/94%)	85.00%±2.75% (80%/89%)	96.32%±1.26% (95.24%/99.13%)	79.45%±2.55% (76.47%/83.53%)
	Class-dependent	99.87%±0.42% (98.67%/100%)	99.87%±0.42% (98.67%/100%)	86.60%±2.72% (83%/90%)	96.41%±0.98% (94.94%/97.97%)	80.16%±3.53% (75.29%/87.06%)
CSM + weighted	Class-dependent	99.73%±0.56% (98.67%/100%)	99.60%±0.72% (98%/100%)	87.10%±2.08% (85%/90%)	96.80%±0.89% (95.67%/98.70%)	81.93%±2.43% (78.82%/85.88%)
RELIEFF + without weighted	class-independent	/	/	83.90%±3.51% (78%/90%)	96.62%±1.10% (95.24%/98.70%)	79.67%±3.23% (76.47%/85.19%)
	class-dependent	/	/	85.60%±2.88% (80%/92%)	96.49%±0.81% (95.67%/98.27%)	80.50%±3.75% (75.29%/85.19%)

TABLE V
TOTAL NUMBERS OF SELECTED FEATURES AND THE CLASSIFICATION ACCURACIES FOR SOAP AND RELIEFF METHODS [49]

Data set	Number of Selected Features for SOAP	Accuracy for SOAP with 1NN	Number of Selected Features for RELIEFF	Accuracy for RELIEFF with 1NN
Waveform40	12.99	79.33%	5.77	73.09%
Segment	7	91.29%	15.04	97.19%
vehicle	1.09	46.50%	5.81	61.28%

TABLE VI
AVERAGE NUMBER OF FEATURES SELECTED BY CLASS-INDEPENDENT AND CLASS-DEPENDENT METHODS WITH THE CSM AND MRMR RANKING MEASURES AND "ONE-AGAINST-ALL" STRATEGY, FOR THE HDR, DNA, AND SCENE DATA SETS

Feature selection	Classes	CSM ranking			mRMR ranking		
		HDR	Scene	DNA	HDR	Scene	DNA
Class-independent	all classes	64.6±5.4	94.4±9.8	46.8±9.9	47.8±4.8	84.5±16.3	40.5±8.9
Class -dependent	class 1	15.5±9.6	38.2±5.1	19±11.5	10.8±3.1	24.6±11.2	18±9.9
	class 2	38.7±7.3	40.4±1.1	37.8±7.5	26.7±13.4	27±7.1	32.3±4.7
	class 3	42.8±8.5	43±3.1	35.4±11.9	10.4±0.5	25.2±10.4	25.5±7.6
	class 4	36.3±11	38±5.1	/	25.4±11.3	26.6±9.9	/
	class 5	36.9±12.6	31.8±9.4	/	10±2.4	15.6±5.3	/
	class 6	35±10.6	44.4±4.9	/	28±14.3	30.1±7.3	/
	class 7	29.1±12.9	/	/	19±7.8	/	/
	class 8	26.1±13.1	/	/	27.2±10.9	/	/
	class 9	40.5±3.7	/	/	15.4 ±6.4	/	/
	class 10	26.7±7.4	/	/	13.2±9.9	/	/
	Average	32.76±3.1	41.93±2.3	33.4±6.2	20.9±3.9	35.8±6.4	28.3±4.9

cial data sets, the *class-dependent* feature selection method produces better results than the *class-independent* feature selection method. Besides, in Table IV, we list the comparison between the results obtained from the proposed general classifier in Fig. 2 and those obtained from the classifier that has not introduced weights to each class-dependent model's output. The proposed general classifier in Fig. 2 consistently produces better results than the classifier without weights on each model's output.

We also list some published results on numbers of selected features with *class-independent* feature selection methods SOAP [49] and RELIEFF [30] with threshold 0.05 and classification accuracies with the 1 nearest-neighbor (1NN) classifier (Table V). Ruiz *et al.* [49] adopted the filter feature selection methods to select features; it had good performance for some data sets. For example, the Ionosphere data set originally had 34 features with an accuracy 86.78%. With the SOAP method, on average, 31.55 features were selected and the accuracy was 87.07%. With the RELIEFF method, on average, 30.88 features were selected with an accuracy 87.49%. However, the performance of those filter feature selection methods was not stable. For example, in the Vehicle data set, the SOAP method had six features selected from original 18 features with an accuracy

TABLE VII
CLASSIFICATION COMPARISONS BETWEEN THE METHODS OF CLASS-INDEPENDENT AND CLASS-DEPENDENT FEATURE SELECTION WITH THE mRMR AND CSM RANKING MEASURES AND "ONE-AGAINST-ALL" STRATEGY, CLASS-INDEPENDENT AND CLASS-DEPENDENT FEATURE EXTRACTION WITH KFDA AND THE "ONE-AGAINST-ALL" STRATEGY, CLASS-DEPENDENT FEATURE SELECTION WITH THE mRMR RANKING MEASURE AND THE "ONE-AGAINST-ONE" STRATEGY FOR THE HDR, DNA, AND SCENE DATA SETS

Feature selection approach		Mean accuracy ± std (minimum/maximum accuracy)		
		HDR	Scene	DNA
Without feature selection		98.13%±0.45% (97.30%/98.70%)	78.67%±1.14% (76.09%/79.35%)	95.89%±0.48% (95.03%/96.54%)
Feature selection (mRMR + weighted)	class-independent	96.93%±0.42% (96.60%/97.80%)	77.30±1.32% (74.99%/78.92%)	95.71%±0.52% (95.03%/96.37%)
	class-dependent +"one-against-all"	98.30%±0.60% (97.40%/98.90%)	78.93%±0.83% (77.56%/79.43%)	96.60%±0.33% (96.04%/96.88%)
	class-dependent +"one-against-one"	97.85%±1.70% (93.50%/99.00%)	73.90%±1.34% (72.16%/77.09%)	96.12%±0.71% (95.19%/97.55%)
Feature Selection (CSM + weighted)	class-independent	93.96%±2.20% (91.90%/96.70%)	73.66%±1.69% (70.57%/75.5%)	95.64%±0.83% (93.76%/96.63%)
	class-dependent +"one-against-all"	97.90%±0.39% (97.20%/98.30%)	78.27%±1.69% (73.66%/79.35%)	96.19%±0.30% (95.95%/96.71%)
Feature extraction (KFDA) (Filters)	class-independent	97.23%±0.42% (96.80%/97.70%)	68.84%±1.31% (65.97%/70.32%)	95.81%±0.46% (95.03%/96.54%)
	class-dependent +"one-against-all"	97.52%±0.27% (97.10%/98.90%)	70.14%±2.28% (68.23%/76.22%)	95.74%±0.39% (95.19%/96.21%)

46.5%, and the RELIEFF method had on average 5.81 features selected with an accuracy 61.28%. Although the number of features was greatly reduced, the classification accuracies were far less than the original accuracy 69.48% without feature selection [49]. In these cases, we believe our wrapper approach with class-dependent feature selection method is better compared to these filter methods, in terms of stable performance of maintaining or improving the classification accuracy.

In Table VII, we make the following comparisons about the classification results: between class-independent and class-dependent feature selection with the CSM and mRMR measures, respectively, using the "one-against-all" strategy; between the "one-against-all" strategy and the "one-against-one" strategy using class-dependent feature selection with the mRMR measure; between class-independent and class-dependent feature extraction using the "one-against-all" strategy; and between feature selection and feature extraction methods. The mRMR ranking measure tends to have better performance than the CSM ranking measure. The results obtained by the class-dependent feature selection using the "one-against-all" strategy are a little better than those obtained by the "one-against-one" strategy. Besides, the "one-against-one" strategy attempts to be computationally more expensive than the "one-against-all" strategy, especially when the number of classes is large. For feature extraction methods, we can see that the class-dependent feature extraction method produces better accuracies than the class-independent feature extraction method for all data sets except the DNA data. Also our proposed class-dependent feature selection method produces better accuracy than the class-dependent feature extraction method, especially for the Scene data with 8.79% increase in accuracy.

IV. CONCLUSION

In this paper, we have proposed a general wrapper approach to class-dependent feature selection. In order to demonstrate the scenario mentioned in the Introduction, we generated two data sets A and B. The experimental results for the two artificial data show that our class-dependent feature selection method can select those specific feature subset for each class, i.e., class-dependent feature subsets, so that the number of noise features or redundant features can be greatly reduced and the classification accuracy is substantially increased. Also the experimental results for other six real data sets show that each feature may indeed have different classification capability in discriminating different classes, and our method has improved or at least maintained the classification accuracy, while reducing the number of features. Hence our general wrapper approach to *class-dependent* feature selection is very effective in deleting irrelevant or redundant features for different classes. The feature subsets that Oh *et al.* [41] obtained are not dependent on a classifier, but in accordance to a predefined dimension. Since the CENPARMI numerical database used in [41] is not publicly available, we did not make a direct comparison between our method and Oh's method [41].

Compared with Baggenstoss' class-specific method [1]–[3], our method has the following differences. First, in selecting the class-dependent feature subsets, we utilized the strategy of "one-against-all" to construct binary classification problems and select class-dependent features, whereas Baggenstoss proposed a "one-against-reference class" strategy, i.e., introducing a reference class [1]–[3] against each class to ensure dimension reduction, because the strategy of "one-against-all" for class-dependent feature selection lacks theoretical basis for dimensionality reduction [2]. Baggenstoss demonstrated an example in the area of signal processing about selecting the correlated Gaussian noise as the reference class [3] and presented mathematical descriptions on how to select the reference class [1], [2]. Baggenstoss' "one-against-reference class" strategy provides theoretical basis for dimensionality reduction, however, some potential assumptions and constraints

[1] are required to determine a reference class. By comparison, the "one-against-all" strategy is easy to be executed in practical applications. For example, both Oh *et al.* [41] and Liu *et al.* [36] adopted the "one-against-all" strategy in their class-specific methods. Second, in the process of designing our general class-dependent classifier, we propose to assign a weight on each model's output for a fair comparison among those class-dependent feature spaces. Here, each weight is the likelihood with which a sample belongs to each class when classifying the sample with a union set of various class-dependent feature subsets. Although this weight approach is a heuristic method, it produces good results in our experiments. As to Baggenstoss' class-specific classifier [2], Baggenstoss proposed the pdf projection theorem, i.e., utilizing a J-function to project those class-specific features' pdfs back into the original data space to obtain a fair comparison. In this way, he also settled the curse of dimensionality for PDF estimation in the low-dimensional class-specific feature space. Although our class-dependent feature selection method has aforementioned differences from Baggenstoss' class-specific method, the proposed heuristic in our method is inspired from Baggenstoss' concept of fair comparison. How to effectively adopt Baggenstoss' pdf projection theorem in our method will be subject of future studies.

Feature selection is challenging because of the tradeoff between time complexity and optimality. The feature selection method proposed in our approach includes two major elements, i.e., class-dependency and wrappers, both of which need more time to seek desirable feature subsets, compared with class-independency and filters. We have shown that wrappers usually ensure higher classification accuracy than filters and class-dependent methods can have more potential to reduce the dimensionality of features than class-independent methods [1]. In order to find the approximately optimal feature subset for each class, various feature ranking measures, wrappers together with various search strategies, e.g., exhaustive search, forward search, and backward search, can be adopted in the framework of our approach. In this paper, the RELIEFF, CSM, and mRMR ranking measures are adopted and partially compared. The more powerful the ranking measure is, the better the selected feature subset and the better the classification performance.

Because of the adoption of ranking measures, we recognize that the approach proposed in this paper is not a pure wrapper. It is a hybrid method between the filter and wrapper [35]. Since we have proposed a framework about a wrapper approach to class-dependent feature selection, we can use the pure wrapper approach in our experiment to replace the proposed hybrid method. Our focus here is on evaluating the advantages of class-dependent feature selection methods over class-independent feature selection methods. We also used a pure wrapper in our general frame work on the Waveform40 data set. The search strategy utilized in the pure wrapper approach is the forward selection search method [45], [48], which begins with an empty set and each time adds one feature until all features are added. At each step, the feature which most increases the performance of the learning machine is included. The pure wrapper approach to class-independent feature selection selected on average 12.2 features with standard deviation 2.2 (i.e., 12.2 ± 2.2) from the original 40 features, and the corresponding classification accuracy is $84.41\%\pm0.63\%$. The pure wrapper approach to class-dependent feature selection selected 7 ± 0, 7 ± 0, 8.3 ± 0.6 features for three classes, respectively, with the classification accuracy $84.12\%\pm0.98\%$. When comparing the numbers of features selected by the pure wrappers with those by the hybrid methods (see Table VI), we can see that the pure wrapper both selected much fewer features than the two hybrid methods, whereas the classification accuracies produced by the pure wrapper approach are lower than those by the two corresponding hybrid methods, especially for the class-dependent feature selection (see Table IV). This may be interpreted as removing more features leading to loss of important information. Moreover, even in the simple forward search strategy [45], [48], the pure wrapper approach is still computationally more expensive than the proposed hybrid method. In fact, Guyon *et al.* [20] and Legrand *et al.* [35] have emphasized that feature ranking based on filters is computationally efficient and statistically robust against overfitting. Therefore, in real applications, although feature ranking based on filters together with sequential forward search may yield suboptimal feature subsets, one may still choose the proposed hybrid techniques so as to take advantages of the filter and wrapper approaches [35], [58]. At the same time, the SVM is chosen as the wrapper due to its good characteristics and originally being a binary classifier. There are other classifiers, e.g., the MLP and RBF, which can also be used in our framework.

Since our feature selection method requires determination of feature subsets of every class, it is likely to be more computationally expensive compared to other class-independent feature selection methods. However, the extra computational cost may be worthwhile in certain applications where improvements of accuracy or reduction of data dimensionality are very important and meaningful.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and three reviewers for their comments and suggestions that helped to significantly improve this paper.

REFERENCES

[1] P. M. Baggenstoss, "Class-specific classifier: Avoiding the curse of dimensionality," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pt. 2, pp. 37–52, Jan. 2004.

[2] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 672–685, Mar. 2003.

[3] P. M. Baggenstoss, "Class-specific features in classification," *IEEE Trans. Signal Process.*, vol. 50, no. 12, pp. 3428–3432, Dec. 2002.

[4] J. Bins and B. Draper, "Feature selection from huge feature sets," in *Proc. Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, 2001, pp. 159–165 [Online]. Available: http://citeseer.ist.psu.edu/bins01feature.html

[5] C. L. Blake and C. J. Merz, "UCI Repository of Machine Learning Databases," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, Tech. Rep., 1998 [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[6] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Proc. Nat. Acad. Sci. USA*, vol. 95, pp. 933–942, 1998.

[7] L. F. Bo, L. Wang, and L. C. Jiao, "Kernel Fisher Discriminant Analysis Toolbox," Inst. Intell. Inf. Process., Xidian Univ., Xi'an, China, Tech. Rep., Mar. 2006.

[8] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Ackinger, P. Simard, and V. Vapnik, "Comparison of classifier methods: A case study in handwriting digit recognition," in *Proc. Int. Conf. Pattern Recognit.*, 1994, pp. 77–87.

[9] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," 2007 [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] L. H. Chen and S. Chang, "An adoptive learning algorithm for principle component analysis," *IEEE Trans. Neural Netw.*, vol. 6, no. 5, pp. 1255–1263, Sep. 1995.

[11] C. Cortes and V. Vapnik, "Support-vector network," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[13] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," *Comput. Learn. Theory*, pp. 35–46, 2000.

[14] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[15] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[16] J. Friedman, "Another approach to polychotomous classification," Dept. Statistics, Stanford Univ., Stanford, CA, 1996 [Online]. Available: http://www-stat.stanford.edu/reports/friedman/poly.ps.Z

[17] X. J. Fu and L. P. Wang, "Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 33, no. 3, pp. 399–409, Jun. 2003.

[18] X. J. Fu and L. P. Wang, "A GA-based novel RBF classifier with class-dependent features," in *Proc. Congr. Evol. Comput.*, 2002, vol. 2, pp. 1890–1894.

[19] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection-theory and algorithms," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, 2004, pp. 43–50.

[20] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[21] S. Halgamuge and L. P. Wang, Eds., *Classification and Clustering for Knowledge Discovery*. Berlin, Germany: Springer-Verlag, 2005.

[22] S. Halgamuge and L. P. Wang, Eds., *Computational Intelligence for Modeling and Predictions*. Berlin, Germany: Springer-Verlag, 2005.

[23] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Waikato Univ., Hamilton, New Zealand, 1998.

[24] C.-W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2003.

[25] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[26] H. G. John, "Enhancements to the data mining process," Ph.D. dissertation, Comput. Sci. Dept., School Eng., Stanford Univ., Stanford, CA, 1997.

[27] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. 10th Nat. Conf. Artif. Intell.*, 1992, pp. 129–134.

[28] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: A stepwise procedure for building and training a neural network," in *Neurocomputing: Algorithms, Architectures and Applications*, ser. NATO ASI, F. Fogelman Soulié and J. Hérault, Eds. New York: Springer-Verlag, 1990, vol. F68.

[29] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 284–292.

[30] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. Eur. Conf. Mach. Learn.*, F. Bergadano and L. de Readt, Eds., Catania, Sicily, Italy, Apr. 1994, pp. 171–182.

[31] U. Krebell, *Pairwise Classification and Support Vector Machines, Advances in Kernel Methods-Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp. 255–268.

[32] P. Langley, *Elements of Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1996.

[33] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *Proc. 10th Conf. Uncertainty Artif. Intell.*, 1994, pp. 399–406.

[34] M. Last, A. Kandel, and O. Maimon, "Information-theoretic algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 22, pp. 799–811, 2001.

[35] G. Legrand and N. Nicoloyannis, "Feature selection method using preferences aggregation," in *Proc. Mach. Learn. Data Mining*, P. Perner and A. Imiya, Eds., 2005, pp. 203–217.

[36] C. L. Liu and H. Sako, "Class-specific feature polynomial classifier for pattern classification and its application to handwritten numerical recognition," *Pattern Recognit.*, vol. 39, no. 4, pp. 669–681, 2006.

[37] Q. S. Liu, H. Q. Lu, and S. D. Ma, "Improving kernel Fisher discriminant analysis for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 42–49, Jan. 2004.

[38] W. Malina, "Two-parameter Fisher criteria," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 31, no. 4, pp. 629–636, Aug. 2001.

[39] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Proc. Neural Networks Signal Process. IX*, 1999, pp. 41–48.

[40] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[41] I. S. Oh, J. S. Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1089–1094, Oct. 1999.

[42] I. S. Oh, J. S. Lee, and C. Y. Suen, "Using class separation for feature analysis and combination of class-dependent features," in *Proc. 14th Int. Conf. Pattern Recognit.*, 1998, vol. 1, pp. 453–455.

[43] H. Peng, F. Long, and C. Ding, "Feature selection based on MI: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[44] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2000, vol. 12, pp. 547–553.

[45] B. Raman and T. R. Ioerger, ""Enhancing learning using feature and example selection," M.S. Thesis, Dept. Comput. Sci., Texas A&M Univ., College Station, TX, 2003.

[46] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, Jul. 2000.

[47] A. L. Rendell and R. Sheshu, "Learning hard concepts through constructive induction: Framework and rationale," *Comput. Intell.*, vol. 6, pp. 247–270, 1990.

[48] J. Reunanen, "Overfitting in making comparisons between variable selection methods," *J. Mach. Learn. Res.*, vol. 3, pp. 1371–1382, 2003.

[49] R. Ruiz, J. S. Aguilar-Ruiz, and J. C. Riquelme, "SOAP: Efficient feature selection of numeric attributes," in *Ibero-Amer. Conf. Artif. Intell. (IBERAMIA)*, 2002, pp. 233–242.

[50] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognit. Lett.*, vol. 10, pp. 335–347, 1989.

[51] The International HapMap Consortium, " Integrating ethics and science in the international HapMap project," *Nature Rev. Genetics*, vol. 5, pp. 467–475, 2004.

[52] M. P. Tu, L. Zhen, and B. A. Russ, "Choosing SNPs using feature selection," in *Proc. IEEE Comput. Syst. Bioinf. Conf.*, 2005, pp. 301–309.

[53] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[54] L. P. Wang, Ed., *Support Vector Machines: Theory and Applications*. New York: Springer-Verlag, 2005.

[55] L. P. Wang and X. J. Fu, *Data Mining With Computational Intelligence*. Berlin, Germany: Springer-Verlag, 2005, p. 276.

[56] J. Weston and C. Watkins, "Multi-class support vector machines," in *Proc. Eur. Symp. Artif. Neural Netw.*, M. Verleysen, Ed., Brussels, Belgium, 1999, pp. 219–224.
[57] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, 2004.
[58] Z. X. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 37, no. 1, pp. 70–76, Feb. 2007.

Lipo Wang received the B.S. degree from National University of Defense Technology, Hunan, China, in 1983 and the Ph.D. degree from Louisiana State University, Baton Rouge, in 1988.

In 1989, he was a Postdoctoral Fellow at Stanford University. In 1990, he was a faculty member in the Department of Electrical Engineering, University College, ADFA, University of New South Wales, Australia. From 1991 to 1993, he was on the staff of the Laboratory of Adaptive Systems, National Institutes of Health, Bethesda, MD. From 1994 to 1997, he was a tenured faculty member in computing at Deakin University, Australia. Since 1998, he has been an Associate Professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is author or coauthor of over 60 journal publications, 12 book chapters, and 90 conference presentations. He holds a U.S. Patent in neural networks. He authored two monographs and edited 16 books.

Dr. Wang is an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS (since 2002), the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION (since 2003), and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (since 2005). He is the Area Editor of the *Soft Computing* journal (since 2002). He serves on the Editorial Board of five additional international journals and was on the Editorial Board of three other journals. He is Vice President–Technical Activities, IEEE Computational Intelligence Society (2006–2007) and served as Chair of Emergent Technologies Technical Committee (2004–2005). He has been on the Governing Board of the Asia-Pacific Neural Network Assembly since 1999 and served as its President in 2002/2003. He was Founding Chair of both the IEEE Engineering in Medicine and Biology Chapter Singapore and the IEEE Computational Intelligence Chapter Singapore. He serves/served as General/Program Chair for 11 international conferences and as member of steering/advisory/organizing/program committees of over 110 international conferences. He was keynote/panel speaker for several international conferences.

Nina Zhou received the B.S. degree in electrical engineering from Wuhan University of Science and Technology, Wuhan, China, in 2000 and the M.S. degree of electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2004. Since 2004, she has been working towards the Ph.D. degree in electrical and electronical engineering (EEE) at the Nanyang Technological University (NTU), Singapore.

Since 2007, she has been working as a Research Associate at the Center for Signal Processing, NTU. Her research interests are in the areas of bioinformatics data mining, feature selection, and classification on biomedical data and audio signals.

Feng Chu received the B.Eng. degree from Zhejiang University, Hangzhou, China and the M.Eng degree from Huazhong University of Science and Technology, Wuhan, China in 1995 and 2002, respectively. Since 2002, he has been working towards the Ph.D. degree at the Nanyang Technological University, Singapore.

Since 2005, he has also worked for Siemens Pte Ltd. and subsequently for Infineon Technologies Asia Pacific Ltd. in Singapore as a Senior Research and Development Engineer. His research interests include: computational intelligence, data mining, and their applications, e.g., bioinformatics, computational finance, etc.