Feature Subset Selection by SVM Ensemble

Tao Ban*, Daisuke Inoue*,
*National Institute of Information and Communications Technology,
Koganei, Tokyo 184-8795, Japan
Email: bantao@nict.go.jp; dai@nict.go.jp

Abstract—**Feature selection (FS) has proven to be useful to improve the generalization performance of classifiers. For applications with a small number of instances but a large number of input features, FS methods based on single classifier evaluation are subject to instability. We propose a new FS algorithm based on SVM ensemble learning. First, an ensemble of SVM classifiers are trained with re-sampled subsets of the training data. Then, with a predefined feature ranking criterion, a new stability criterion is defined on the ranking criterion values among the classifiers to measure the relevance of a certain feature. This measure favors the features which have stable ranking criterion values over the features whose ranking criterion values are subject to large variations. The unstable features usually do not have much relative information to the class label, and can be removed to improve the generalization performance of the classifier. To rank the features, the method only requires a small number of SVM classifiers to be trained. It is very fast to solve feature selection problems with a large number of input features. Combined with a backward elimination procedure, this method is robust to feature selection problems with very small sample sizes. In this paper, we evaluate its performance on nonlinear selection tasks.**

I. Introduction

Feature selection has attracted extensive research attention during the past few decades [1], [2], [3]. In classification problems, the goal of feature selection is to eliminate irrelevant variables to enhance the generalization performance of a given learning algorithm. Especially in application areas where available datasets are often with thousands of input features, feature selection can help to: (1) defy the curse of dimensionality and improve the generalization performance; (2) speed up computation in training and prediction; (3) reduce data gathering and storage cost; (4) facilitate data visualization and lead to some insights into the concept to be learned.

In addition to the traditional feature selection methods which usually fall in the categories of wrapper methods and filter methods [4], some novel feature selection methods based on Support Vector Machines (SVMs) are recently proposed [5], [6] and have shown their advantages for real-world applications. Generally speaking, preceding feature selection is still an essential step to help design SVM classifiers with high generalization ability in application areas such as gene expression analysis and bio-informatics, where SVM classifiers suffer from numerous redundant and noisy features. The SVM based feature selection methods can be classified into two categories: backward or forward feature selection based on some predefined selection criteria [5], [6]; and embedded SVM feature selection, in which a feature selection criterion is added to the objective function [7], [8].

In this paper, we discuss feature selection for unstable classifiers. A classifier is called unstable if small changes in the training data lead to significantly different classifiers and relatively large changes in accuracy [9]. Generally, the larger the number of input features and the smaller the number of training samples the more difficult to train a stable classifier. Independent from classifier design, there are basically two approaches to improve a given classifier's generalization ability. On one hand, feature selection methods can remove noisy and redundant features and get an improvement in generalization. On the other hand, ensemble learning methods such as Bagging [10], Boosting [11], and AdaBoost [12], can help improve the generalization performance of an unstable classifier in a statistical sense. Similar idea on deploy ensemble learning to improve the generalization performance of learning system can found in methods such as Random Forests (RF) [13], [14], [15] and other [16], [17].

The SVM classifiers trained in the SVM-based feature selection methods, e.g. the SVM Recursive Elimination (SVM-RFE) method in [5], may overfit to the training set and lead to failure in identifying the feature subset with best prediction power. Instead of doing feature selection base on the evaluation of features with a single classifier, we propose to employ ensemble learning in the process. Suppose we are given a sequence of training sets $\mathcal{D}_j$ $(j = 1, \ldots, J)$, each of which consists of ℓ_j independent observations from the same underlying distribution. Our goal is to use $\mathcal{D}_j$ to select a more reliable subset of features than that acquired by a single set $\mathcal{D} = \bigcup_{j=1}^{J} \mathcal{D}_j$. Borrowing the idea from Bagging, we can first train an ensemble of SVM classifiers from the datasets. Then, the relevance of a certain feature is estimated by its contribution to the class margins in an ensemble of SVMs. Features which contribute much to the majority of the SVMs are selected as significant features, while others are eliminated from the feature set. Because the statistical property of each feature can be obtained from an ensemble of SVMs simultaneously and can be used to rank the features, only a few number of SVMs need to be trained for feature selection. Compared with the backward or forward selection, this method is very fast for a feature selection task with large number of input features. Furthermore, we can combine the proposed method with a backward selection procedure to improve its stability.

The rest of the paper is organized as follows. Section II gives a brief review on SVMs and the SVM-RFE algorithm. Section III details the proposed algorithm. Numerical experiments on toy problems and real-world datasets are described in Section IV. Some discussions on the algorithms are reported in Section V. Section VI concludes the paper.

II. SVMs and SVM Based Feature Selection

In this section, we first briefly review the formulation of support vector learning, then we explain commonly used SVM-based feature ranking criteria and show their applications in the SVM-RFE algorithm.

A. Support Vector Learning

Support Vector Machines [18] realize the following idea. Suppose we are given two classes of samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)|\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}, i = 1, \cdots, \ell, \}$ as a training set. We first map the input vector $\boldsymbol{x}_i$ into a high (possibly infinite) dimensional feature space, F, through a nonlinear mapping function Φ; then construct the optimal hyperplane that realizes the maximal margin in this space. With the so called *kernel trick*, the mapping Φ is implicitly implemented by some kernel function $K(\cdot, \cdot)$, which defines an inner product in the feature space. The decision function given by SVM is thus in a linear form of $\Phi(\boldsymbol{x})$ as:

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b, \tag{1}$$

where $\boldsymbol{w}$ is the normal vector of the decision hyperplane in F and b the bias. A novel sample $\boldsymbol{x}$ with $f(\boldsymbol{x}) > 0$ is assigned to class {1}, otherwise it is assigned to class {-1}.

For an SVM classifier with misclassified samples being linearly penalized with a positive soft margin parameter C, the optimization problem can be written as:

$$\left\{ \begin{array}{ll} \min & \frac{1}{2}\|\boldsymbol{w}\|^2 + \ C \sum\limits_{i=1}^{\ell} \xi_i, \\ s.t. & y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \cdots, \ell, \end{array} \right. \tag{2}$$

where the nonnegative slack variables ξ_i, $i = 1, \cdots, \ell$, are introduced to guarantee feasible solutions always exist. The solution of the problem in Equation (2) can be obtained with the Lagrangian theory and $\boldsymbol{w}$ can be derived as:

$$\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i^* y_i \Phi(\boldsymbol{x}_i), \tag{3}$$

where α_i^* is the solution of the following quadratic optimization problem, usually called a *dual problem*:

$$\left\{ \begin{array}{ll} \max & W(\boldsymbol{\alpha}) = \sum\limits_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), \\ s.t. & \sum\limits_{i=1}^{\ell} y_i \alpha_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, \ell. \end{array} \right. \tag{4}$$

B. Ranking Criteria for Feature Selection

SVMs provides us many statistics to estimate their generalization performance from bounds on the leave-one-out error L. The leave-one-out error is known to be an unbiased estimator of the generalization performance of a classifier trained on $\ell - 1$ examples. One of the most common L error bounds for SVMs is the radius/margin bound (for decision function with non-zero bias b) [18]:

$$L \leq 4r^2 \|\boldsymbol{w}\|^2, \tag{5}$$

where r is the radius of the smallest hypersphere that contains all the mapped data $\Phi(\boldsymbol{x}_i)$. The geometrical margin δ of a separating hyperplane, which is the distance between the hyperplane and an *unbounded support vector*, can be obtained by

$$\delta = 1/\|\boldsymbol{w}\|. \tag{6}$$

Thus, for an SVM, maximizing the margin corresponds to minimizing $\|\boldsymbol{w}\|$.

Given the optimal solution of Equation (4) as $\boldsymbol{\alpha}^*$ and $\boldsymbol{w}^*$, it is easy to check that,

$$\|\boldsymbol{w}^*\|^2 = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{7}$$

For a linear SVM, Equation (7) can be simplified to

$$\boldsymbol{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* \boldsymbol{x}_i. \tag{8}$$

For the linear case, we can see that if some elements of $\boldsymbol{w}^*$ are zero, the deletion of the associated input features will not lead to variation in the decision function. Furthermore, a feature associated with an element near to zero in $\boldsymbol{w}^*$ may be considered insignificant, and can probably be deleted without degeneration in generalization ability. Thus, the ranking criterion of the kth feature for a linear problem is defined as

$$R_k = \sqrt{\|\boldsymbol{w}^*\|^2 - \|\boldsymbol{w}^{*(k)}\|^2} = |\sum_{i=1}^{\ell} y_i \alpha_i^* x_{ik}|, \tag{9}$$

where x_{ik} is the kth element of $\boldsymbol{x}_i$, and $\boldsymbol{w}^{*(k)}$ is obtained from $\boldsymbol{w}$ by setting all components x_{ik} to 0 for $i = 1, \ldots, \ell$.

We can extend this discussion to the nonlinear case where deletion of an input feature corresponds to deletion of multiple features in the feature space. In this case, features which contribute least to $\|\boldsymbol{w}^*\|$ in Equation (7) can be possible candidates for deletion. The contribution of the kth feature to $\|\boldsymbol{w}^*\|$ can be evaluated as

$$\begin{aligned} R_k &= \sqrt{\|\boldsymbol{w}^*\|^2 - \|\boldsymbol{w}^{*(k)}\|^2} \qquad (10) \\ &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* (K(\boldsymbol{x}_i, \boldsymbol{x}_j) - K(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(k)}))^{1/2}, \end{aligned}$$

where $\boldsymbol{x}^{(k)}$ is the vector with the kth feature of $\boldsymbol{x}$ set to 0. Note that for the sake of simplicity and speedup of computation, $\boldsymbol{\alpha}^{*(k)}$, the solution of the optimization problem with the kth feature deleted, is supposed to be equal to $\boldsymbol{\alpha}^*$.

Another similar ranking criterion related with SVM has been proposed in [19]. The idea was that from Equation (7), $||w||^2$ can be viewed as a function of the $d \times \ell$ real variables x_{ik}, $i = 1, \ldots, \ell$, $k = 1, \ldots, d$. Based on the the assumption that $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^{*(k)}$, they compute the partial derivatives of $||w||^2$ with respect to all x_{ij}, and assign to the kth feature the score

$$V_k = \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* \frac{\partial K(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial x_{jk}} \tag{11}$$

which gives the credit of the feature. Here, x_{jk} is the kth component of $\boldsymbol{x}_j$ and the partial derivative is evaluated at $\boldsymbol{x}_i$.

For linear kernels, Equation (11) is equivalent to Equation (9) while for nonlinear kernels, it is different from Equation (11) which is adopted in our experiments.

C. SVM-RFE Algorithm

The SVM-RFE algorithm [5] was proposed to select relevant genes for cancer classification problems. It follows the backward selection method: one starts with all the features and removes one or a subset of features at a time until a predefined number of features are left. The removed feature is the one whose removal minimizes the variation of $\|\boldsymbol{w}^*\|^2$, i.e. the kth feature with smallest R_k is eliminated.

Step 1	Initialization: P_f = $\{1,...,d\}$.
Step 2	Loop for feature selection
	1^o Train an SVM classifier with the features in P_f;
	2^o Compute R_k, ($k = 1, ..., \|P_\mathrm{f}\|$);
	3^o E_f = arg $\min_k$ R_k; (E_f can be multiple features.)
	4^o $P_\mathrm{f} = P_\mathrm{f} - E_\mathrm{f}$;
Step 3	If $\|P_\mathrm{f}\| \geq r$, then go to step 2, otherwise stop.

TABLE I. THE BACKWARD SELECTION ALGORITHM FOR SVM-RFE. IN THE TABLE, P_f IS THE SET OF PRESERVED FEATURES FOR SVM TRAINING, E_f A SET OF FEATURES TO BE DELETED, R_k THE RANKING CRITERION FOR THE kTH FEATURE WITH THE CURRENT SVM, r A PREDEFINED NUMBER OF FEATURES, AND $|P_\mathrm{f}|$ IS THE CARDINAL NUMBER OF FEATURE SET P_f.

To search the best r features, a greedy algorithm based on backward selection is performed. A backward sequential selection is used because of its lower computational complexity compared to randomized or exponential algorithms and its optimality in the subset selection problem [20]. Hence, the algorithm starts with all features and repeatedly removes a feature until r features are left or all variables have been ranked. See Table I for the detailed algorithm.

III. Feature Subset Selection by SVM Ensemble

In this section we discuss the proposed feature selection method. Because the features are ranked by some stability evaluation criterion, we refer to the method as *Stability Evaluation* based feature selection, or in short SVM-SE. First we discuss how to generalize the ensemble learning idea to feature selection. Then the stability criterion based on the ranking criterion adopted from SVM-RFE is defined. The SVM-SE method and the combined method with a backward selection procedure are described in Section 3.3.

A. Ensemble Learning Generalized to Feature Selection

Compared with other methods, the SVM-RFE method has shown high performance in solving feature selection problems with microarray datasets. One of the drawbacks of the microarray datasets is that they are always short in number and suffer from abundant noisy features. These problems are usually unstable: a small variation of the input can influence the output of the system greatly. In such a case, the SVM classifiers trained in the ranking procedure of the SVM-RFE method may overfit to the training set and thus lead to failure in identifying the feature subset with best prediction power.

The so called "Bagging"—a name derived from "bootstrap aggregation"—tries to improve the recognition of a given classifier by using multiple versions of a training set, each created by drawing $N < \ell$ samples from $\mathcal{D}$ with replacement [10]. An original classifier tries to learn from a training set $\mathcal{D} = \{(y_i, \boldsymbol{x}_i), i = 1, \ldots, \ell\}$ and get a predictor $\psi(\boldsymbol{x}, \mathcal{D})$. If the input is $\boldsymbol{x}$, the prediction y is given by $\psi(\boldsymbol{x}, \mathcal{D})$. Bagging suggests a better predictor by learning from multiple versions of the training set. Suppose we are given a sequence of learning sets $\mathcal{D}_j = \{(y_i, \boldsymbol{x}_i)\}$ each of which consists of N independent observations from the same underlying distribution as $\mathcal{D}$. Note the averaged predictor of $\psi(\boldsymbol{x}, \mathcal{D}_j)$ over j as $\psi_\mathrm{A}(\boldsymbol{x}) = E_\mathcal{D}\psi(\boldsymbol{x}, \mathcal{D}_j)$, where $E_\mathcal{D}$ denotes the expectation over $\mathcal{D}$, and the subscript A in ψ_A denotes aggregation. $\psi_\mathrm{A}(\boldsymbol{x})$ is able to show better generalization performance on unstable classifiers [10]. For cases where only a single training set $\mathcal{D}$ is available, Bagging uses an imitation of the process by taking repeated bootstrap samples $\mathcal{D}_j$ from $\mathcal{D}$, and form $\psi(\boldsymbol{x}, \mathcal{D}_j)$. The bootstrap samples $\mathcal{D}_j$ are drawn at random, but with replacement. Each $(y_i, \boldsymbol{x}_i)$ may appear repeated times or not at all in any particular $\mathcal{D}_j$. The final prediction is based on the vote of all the predictors $\psi(\boldsymbol{x}, \mathcal{D}_j)$.

We can generalize the idea in Bagging to feature selection problems. Consider the following problem as an example. Suppose we have a sequence of training sets $\mathcal{D}_j = \{(y_i, \boldsymbol{x}_i), i = 1, \ldots, N\}$ independently and identically sampled from a given problem. If we perform feature selection for each of the datasets separately, will the multiple runs of the same feature selection algorithm on different datasets yield identical subsets of features? Generally speaking, if the number of samples in the datasets are sufficient to learn the concept generating the data, the answer could be yes. However, as a dataset (take the example of an microarray dataset) may contain thousands of features but only a few number of instances, the answer probably will be negative. A further question on the above issue can lead us to the proposed method. That is, can we combine the training sets to improve feature selection? Apparently, the answer is yes, for the reason that more data will certainly offer more insight into the concept to be learned. Suppose that all available training data still are not sufficient to get a stable classifier. Then, rather than combining the datasets into one and following the feature selection procedure of SVM-RFE, we would more likely to trust in the performance of the features in an ensemble of classifiers.

B. Stability Evaluation

Given a predefined ranking criterion, the proposed feature selection method can be formally stated as follows.

Suppose we are given a sequence of training sets $\mathcal{D}_j$ ($j = 1, \ldots, J$), each of which consists of ℓ_j independent observations from the same underlying distribution. Our goal is to use $\mathcal{D}_j$ to select a more reliable subset of features than that acquired by a single set $\mathcal{D} = \bigcup_{j=1}^{J} \mathcal{D}_j$. In case when only a single training set $\mathcal{D}$ is available without the replicates, we can follow the procedure in Bagging to resample subsets for training. Let p be the fixed ratio of of the subsets to the full training set. Repeated bootstrap samples $\mathcal{D}_j$ of size $p\ell$ are drawn randomly from $\mathcal{D}$ with replacement. Then $\mathcal{D}_j$ are replicate training sets approximating the distribution underlying $\mathcal{D}$.

From the training sets $\mathcal{D}_j$ ($j = 1, \ldots, J,$), we can learn J SVMs each of which gives an estimation of the separating hyperplane. Given a predefined ranking criterion, each SVM suggests a feature ranking order. Instead of ranking the features by the absolute values of the ranking criterion, we rank them by some stability evaluation. In a statistical sense, the standard deviation of the ranking criterion defined in Equations (9) and (11) may offer a good estimate of the feature's stability. On the other hand, we have to take the absolute value of the criterion into consideration. Thus based on the ranking criteria described in Section 2, we define the *stability criterion* of the kth feature as

$$S_k = |\mu_k(R)| / \sigma_k(R), \tag{12}$$

where $\mu_k(R)$ is the mean of the feature's ranking criterion, $\sigma_k(R)$ the standard deviation. $\mu_k(R)$ and $\sigma_k(R)$ are functions of R. In the rest of the paper, if their is no confusion, we note them as μ_k and σ_k. Then, the larger the absolute mean or the smaller the standard deviation, the more relevant the feature is.

Equation (12) implies that a feature that performs consistently (i.e. has large $|\mu_k|$ but a very small σ_k) will always be ranked better than a feature that is sometimes good and sometimes bad (i.e. has lower $|\mu_k|$ but higher σ_k). Take the problem in Fig.IV-C1 as an example again. For the first feature, $|\mu_1| = 13.937$ and $\sigma_1 = 1.228$. For the second feature, $|\mu_2| = 5.000$ and $\sigma_2 = 2.042$. As we have mentioned, feature 2 offers no discriminability. Still, it has a rather large value of $|\mu_2|$. On the other hand, since it contains mere noise, its ranking criterion has a large variance. The stability criteria of the two features are $S_1 = 11.346$ and $S_2 = 2.449$. In this case, compared with using merely the $|\mu_j|$ or σ_j as the ranking criterion, the proposed stability criterion is more reliable. Generally, when the sign of w_k associated with feature k varies in the SVM ensemble, the stability criterion of the feature will have a small value.

To estimate the standard deviation and rank the features, an ensemble of J SVM classifiers should be trained. Compared to the number of SVMs to be trained for a backward selection algorithm such as SVM-RFE, the number of SVM classifiers to be trained can be reduced greatly: given the number of input features as d, if one feature is removed at a time, SVM-RFE needs to train d SVMs. Generally, $J << d$ holds for datasets with large number of input features, especially for nowaday microarray datasets. However, there are two drawbacks of ranking the features directly on the defined stability criterion. First, since the initialization of the ensemble of SVMs involves a random procedure and the prediction power of the feature subset is sensitive to slight variation in the feature ranking sequence, the test performance varies slightly from time to time. Second, the SVM classifiers with most input features as noise may be greatly different from those trained with less noisy features. Hence, to improve the stability of the algorithm, we can combine the above method with a backward selection procedure. As the experiments show, we can eliminate a rather large number of insignificant features at a step without large influence on the finally preserved features.

C. Algorithm

Here we detail the algorithm of the proposed method. As the features are ranked based on their stability evaluations, the method is called a *Stability Evaluation* based feature selection method (SVM-SE). The procedure is listed in Table II.

Step 1	Initialize J training sets $\mathcal{D}_j$;
Step 2	Loop for $j = 1, \ldots, J$
	1^o Train the SVM classifier for $\mathcal{D}_j$;
	2^o Compute ranking criterion R_k for feature k, $k = 1, \ldots, d$, and store them in an array $\mathcal{R}$;
Step 3	Compute stability criterion S_k ($k = 1, \ldots, d$) from $\mathcal{R}$.
Step 4	Sort the features based on the stability criteria.

TABLE II. THE SVM-SE ALGORITHM. IN THE TABLE, J IS THE NUMBER OF RESAMPLED TRAINING SETS, d THE NUMBER OF INPUT FEATURES.

In Step 1, to initialize the training sets $\mathcal{D}_j$, data are randomly sampled from the training set with replacement. The ratio of a subset to the full training set is denoted by a parameter p. In the loop, K SVM classifiers are trained for all the training sets and for each SVM, the ranking criteria R for all the features are stored in $\mathcal{R}$. Thus, $\mathcal{R}$ is an $d \times J$ array with $\mathcal{R}_{kj}$ storing the ranking criterion of the kth feature computed from the jth training set. In Step 3, the stability criterion S_k for each feature is computed from $\mathcal{R}$. Finally, the features are ranked in a list according to their stability criteria in descending order. The more relevant a feature is to the classification problem, the smaller its index in the list. If the number of features to predict the class label of incoming samples, r, is predefined, the first r features can be used to train an SVM classifier on the full training set and test on new coming data. Otherwise, a simple forward selection procedure based on the ranking order in the list can help to decide how many features are needed to train a classifier with a good generalization ability. Note that, to save memory, during the computation of the stability criterion, only the aggregated sums $\sum_j \mathcal{R}_{jk}$ and $\sum_j \mathcal{R}_{jk}^2$ need to be stored, not the full matrix $\mathcal{R}$.

Generally speaking, we can apply this method to various ranking criteria, for example, the margin based criteria, some span estimation based criteria, or their gradients. As reported by [6], the zero order of $\|\boldsymbol{w}^*\|^2$ criterion used in SVM-RFE, i.e. the margin based criteria defined in Equations (9) and (11), outperforms other criteria with most of the examined datasets. In the experiments, for a linear problem, the criterion in Equation (9) is adopted, while for a nonlinear problem, the criterion in Equation (10) is evaluated.

In the SVM-SE algorithm, only K SVM classifiers need to be trained. Here we take the assumption that these SVM classifiers can give good generalization performance with all the input features. However, as the presence of many noisy features, to make the classifiers more stable, a more feasible method is to perform the procedure for many times, with a subset of features deleted at each time. Thus we can combine the SVM-SE method with a backward selection procedure. To terminate the algorithm, we check the cross validation rate v. Note the recognition rate of the SVM classifier trained from dataset $\mathcal{D}_j$ and tested on $\mathcal{D} - \mathcal{D}_j$ as v_j. v is defined as the mean of v_j, $j = 1, \ldots, J$. If the cross validation rate successively decreases for s times, then we stop the algorithm. The output feature subset is the one with the best cross validation rate.

In Table III, we detail the SVM-SE algorithm with a combined backward selection procedure. In the initialization step, the set of preserved features, P_{f}, is initialized to include

```
Step 1  Initialization: $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell)\}$, $P_{\mathrm{f}} \leftarrow \{1, ..., d\}$,
            $t \leftarrow 0$, $v_{\mathrm{old}} \leftarrow 0.0$, $s \leftarrow s_0$;
Step 2  While $t < s$ and $P_{\mathrm{f}} \neq \phi$
            $1^o$ Compute the stability criteria $S_k$, for $k = 1, \ldots, d$;
            $2^o$ $E_{\mathrm{f}} \leftarrow \arg\min_k S_k$; ($E_{\mathrm{f}}$ can be multiple features.)
            $3^o$ $P_{\mathrm{f}} \leftarrow P_{\mathrm{f}} - E_{\mathrm{f}}$;
            $4^o$ Compute $v_j$ for $j = 1, \ldots, J$;
            $5^o$ $v \leftarrow 1/J \sum_{j=1}^{J} v_j$;
            $6^o$ If $v < v_{\mathrm{old}}$ then $t \leftarrow t + 1$, else $v_{\mathrm{old}} \leftarrow v$, $t \leftarrow 0$;
Step 3  End.
```

TABLE III. BACKWARD FEATURE SELECTION WITH STABILITY EVALUATION. IN THE TABLE, P_{f} IS THE SET OF PRESERVED FEATURES FOR SVM ENSEMBLE TRAINING, E_{f} A SET OF FEATURES TO BE DELETED, S_k THE STABILITY CRITERION FOR THE kTH FEATURE, AND s_0 A PREDEFINED NUMBER OF ITERATIONS.

all the features. In the second step, we use the SVM-SE method in Table II to get the stability criteria S_k for all the features. To speed up the algorithm, multiple features at the end of the ranked list are deleted from set P_{f} in each step. In the experiment, we adopt the following approach: a fixed proportion (noted as a ratio parameter β) of the least relevant features are deleted from the preserved feature set P_{f} at a time. Then the generalization performance of the preserved features in P_{f} is tested by cross validation. During the loop, if the cross validation rate successively decreases for s times, then the algorithm stops. The algorithm also can be stopped when all features are ranked.

IV. EXPERIMENTS

In this section, we report some experimental results with artificial and real-world datasets. We compare the classification performance associated with the selected features acquired by the SVM-RFE method, the proposed SVM-SE method, and the combined method. As references, the recognition rates of stand-alone SVM classifiers are reported.

A. Datasets

Table IV lists the number of input features, training set size, and test set size the datasets reported in the experiments. (The hyperparamters for SVM learning listed in the same table are discussed later.) The first dataset is a toy problem and the rest are real-word tasks.

1) Toy Problem: For the toy experiment, we use the dataset described in [8].

Two features of 52 are relevant. The probabilities of $y = 1$ and $y = -1$ are equal. For $y = -1$, x_1 and x_2 are drawn from $N(\boldsymbol{\mu}_1, \Sigma)$ or $N(\boldsymbol{\mu}_2, \Sigma)$ with equal probability, $\boldsymbol{\mu}_1 = (-3/4, -3)$, $\boldsymbol{\mu}_2 = (3/4, 3)$, and $\Sigma = I$. For $y = 1$, x_1 and x_2 are drawn from two normal distributions with equal probabilities, with $\boldsymbol{\mu}_1 = (3, -3)$, $\boldsymbol{\mu}_2 = (-3, 3)$, and $\Sigma = I$. Here $N(\boldsymbol{\mu}, \Sigma)$ is a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. The remaining features are noises $x_i = N(0, 20)$, $i = 3, \cdots, 52$.

2) Nonlinear Real-life Datasets: For nonlinear feature selection we show results on two datasets: Wisconsin diagnostic breast cancer (WDBC) [21] dataset and the USPS dataset [22]. The USPS dataset contains 10 classes denoting 10 handwritten digits, however as we only consider 2-class problems in the paper, the dataset is divided into pairwise subproblems and two representative subproblems from the 45 are reported.

Dataset	#Train	#Test	#Feature	C	γ
Nonlinear Toy	50	1000	52	100	1
WDBC	200	369	30	100	0.033
USPS: 3 vs. 5	236	1214	256	10	0.031
USPS: 6 vs. 8	100	153	256	10	0.031

TABLE IV. DATASETS IN THE EXPERIMENTS. IN THE TABLE, #TRAIN STANDS FOR TRAINING SET SIZE, #TEST FOR TEST SET SIZE, AND #FEATURE FOR NUMBER OF INPUTS. C IS THE SOFT MARGIN PARAMETER IN THE SVM FORMULATION. γ IS THE WIDTH PARAMETER WHEN A GAUSSIAN KERNEL IS ENGAGED.

B. Computations

Experiments show that a J value ranging from 20 to 100 does not lead to a large variance in the performance. Thus to save computations, we set $J = 20$ in all the experiments. In the SVM-SE algorithm, the features are ranked based on the stability criteria with the J SVM classifiers. Each resampled training set contains 80% of the full training set, e.g. $p = 0.8$.

To get the referential result of the SVM-RFE algorithm, as suggested in [8], one feature is removed in each step. For the proposed combined method, at each iteration, 5% of the features are deleted, i.e. $\beta = 0.05$. To compare with the result of SVM-RFE, we do not stop the algorithm until all features are ranked. Gaussian kernels are used for the nonlinear problems. The hyperparameters C and γ (when a Gaussian kernel is engaged) are selected by 10-fold cross validation on the training set. We list the parameters adopted in the experiment in Table IV. A C parameter ranging from 1 to 10000 leads to the same recognition rate in cross validation. We set $C = 1000$ in case the problem turns to be non-separable with some feature subsets. All the SVM classifiers are implemented with the LIBSVM toolbox [23]. Experiments are run on a Linux server with quad core 3.4GHz Xeon CPU.

As a reference, we also check the performance of a simple correlation coefficients based feature ranking method adapted from [24]. The correlation coefficients of the kth feature is defined as follows:

$$w_k = (\mu_k(+) - \mu_k(-))/(\sigma_k(+) + \sigma_k(-)), \qquad (13)$$

where $\mu_k(\cdot)$ and $\sigma_k(\cdot)$, $k = 1, \ldots, d$, are the mean and standard deviation of the kth feature for the corresponding class. Large positive w_k values indicate strong correlation with class $(+)$ whereas large negative w_k values indicate strong correlation with class $(-)$. We refer to it as a baseline method in the rest of the paper.

C. Numerical Results

1) Artificial Dataset: The recognition rates of the three methods on 1000 test samples for the nonlinear toy problem is listed in Table V. The recognition rate of an SVM trained from the full feature set is shown as a reference. In the table, a recognition rate on the test set is followed by the selected number of features with the best prediction power. For the nonlinear toy dataset, SVM performs poorly: it gives a recognition rate of 0.672. The three methods select the same first 2 features with recognition rate 0.965. As the selected number of features increases, the three methods show comparable performance. The baseline method fails to select the most relevant features of the nonlinear problem.

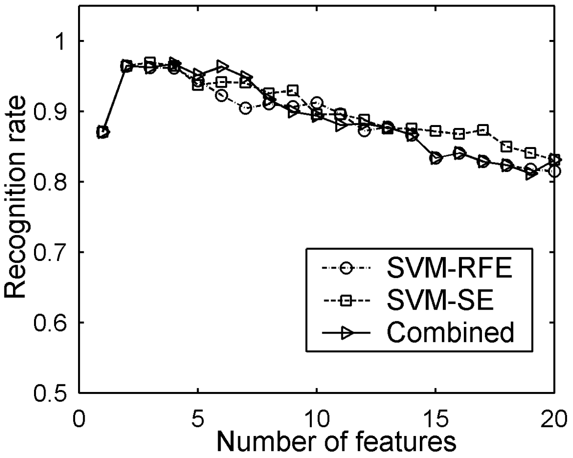


Fig. 1. Feature selection results of the nonlinear toy problem.

2) Nonlinear Real-life Datasets: For the WDBC dataset, we can see from Fig.2(a) that there are many redundant features. All the three SVM based methods can find a relatively small feature subset without too much degeneration in generalization ability: with 5 features, they all achieve a recognition rate above 0.97 — better than the recognition rate, 0.968, yielded by an SVM classifier with all the input features. Obviously, the proposed methods can successfully select the relevant features. The optimal recognition rates given by the four methods are reported in Table V. With about 20 features, the three SVM based methods can give a recognition rate above 0.980. The baseline method gives a recognition rate of 0.976 with 26 features.

In Figures 2(b) and 2(c), we show the results for the two USPS 2-class subproblems. The first subproblem consists of hand written "3" and "5", and the second subproblem consists of "6" and "8". For both of the problems, with about 40 features, we can get SVM classifiers with equivalent recognition rates to SVMs trained with all the input features. With some more features added to the feature subset, the generalization performance can be improved.

As we can read from Table V, SVM-RFE, SVM-SE, and the combined method show comparable performance. For the first subproblem, they can get a recognition rate close to 0.980 with about 50 features, while the baseline method has a recognition rate of 0.957 with 94 features. For the second subproblem, they give almost the same recognition rate with 39, 40, 42 features respectively, while the baseline method has a recognition rate of 0.994 with 98 features. The baseline method also can help to reduce the number of relevant features, however, it works no better than the three SVM based methods.

Figure 3 shows the selected features by the four feature selection algorithms for the two USPS subproblems. In the first row of Fig.3, the class mean of digits are shown. In the other two rows, we show the selected subset of features by SVM-RFE, SVM-SE, the combined method, and the baseline method. The ranking order of each pixel is also shown in the figures. From the figures we can see that SVM-RFE, SVM-SE, and the combined method selects the pixels in the same region although the ranking orders of the features are a little different with each other. The baseline method mainly selects features where the two target classes do not overlap and neglects the overlapping regions. However, the overlapping regions may also carry class information as the three SVM-based methods show.

V. Discussions

In this section, we discuss some questions that have arisen from the proposed algorithm.

A. Computation Costs

SVM based feature selection methods spend most of the running time on SVM learning. Then, the computation cost can be measured by the number of SVM classifiers trained in the procedure.

Let the number of input features be d. For the SVM-RFE method, if one feature is removed at a time, d SVMs should be trained. The number of SVMs to be trained is $O(d)$. For the SVM-SE algorithm, J SVMs have to be trained to get the stability evaluation of the features. Given J as a predefined constant, $O(1)$ SVM classifiers must be trained. For the combined method, as a proportion β of the features are removed at a time, we have to train $-J\frac{log(d)}{log(1-\beta)}$ SVMs. If $J << d$, then the SVM-SE method is expected to outperform SVM-RFE. When the number of input features are large enough the combined method will take less time than SVM-RFE. In practice, SVM-RFE can be sped up by eliminating multiple features at a step, although perhaps not as many as the combined method. Note that as the selected number of features decreases, there will be some speedup in SVM training. So the above assertion is not always true.

Table V also lists the computation time on the datasets. For all the datasets, the SVM-SE method outperforms the other two methods in speed. For the microarray datasets with large number of input features, the combined method is faster than SVM-RFE. For WDBC dataset, SVM-RFE runs faster. For the other datasets, the combined method and SVM-RFE show comparable computation time.

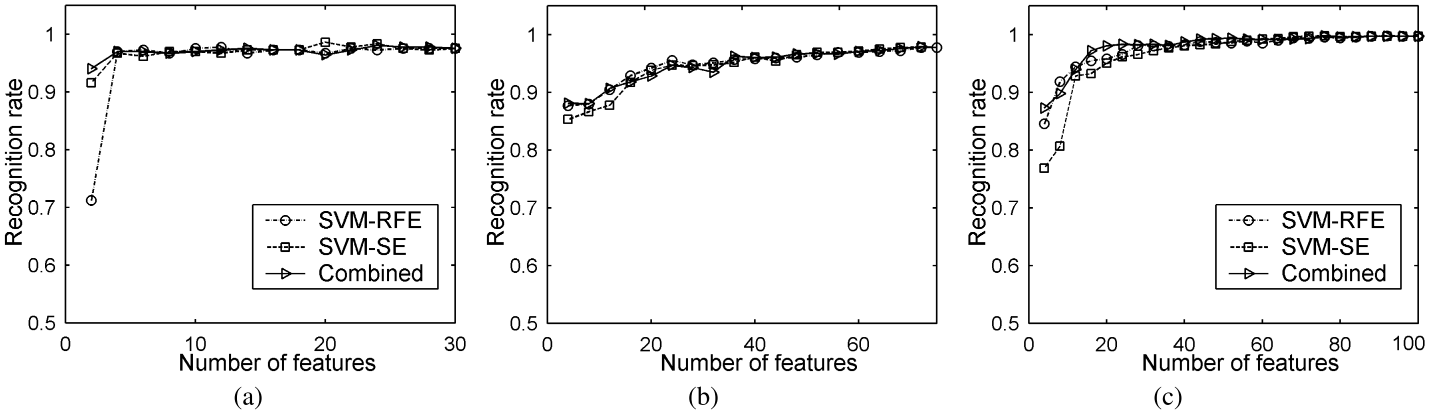
B. Extending to Multi-class Problems

We do not explore application of feature selection to multi-class problems. Here, we suggest an approach to extend a two-class feature selection algorithm to solve multi-class problems. Some general discussions to apply a binary feature selection method to multi-class cases can be found in [25].

SVMs are originally formulated for two-class problems. In the literature, there are many discussions on how to extend SVMs to solve multiclass problems. See [26] for detailed discussions. Among them, the one-against-all SVM and pairwise SVM first divide a multiclass problem into a series of two-class problems and then do classification based on the two-class SVMs according to some voting strategy. Here we suggest to do feature selection using the pairwise SVM technique for the following reasons. (1) Pairwise SVMs are reported to have better performances for real applications. (2) Feature selection can be done more effectively for a pairwise classifier than for

Dataset	SVM	SVM-RFE		SVM-SE		Combined		Baseline	
		Rate	Time	Rate	Time	Rate	Time	Rate	Time
Nonlinear Toy	0.672	0.965(2)	1.05	0.965(2)	0.89	0.965(2)	5.66	0.829(21)	0.42
WDBC	0.968	0.981(25)	0.61	0.986(21)	0.91	0.984(20)	4.89	0.976(26)	0.19
USPS3&5	0.951	0.974(50)	2279	0.974(50)	267.6	0.979(46)	2398	0.957(94)	245.0
USPS6&8	0.985	0.996(39)	987.4	0.997(40)	109.4	0.997(42)	1065	0.994(98)	119.3

TABLE V. FEATURE SELECTION RESULTS ON THE TEST SETS. THE NUMBERS IN BRACKETS SHOWS THE NUMBER OF FEATURES WITH THE OPTIMAL PREDICTION PERFORMANCE.



(a) (b) (c)

Fig. 2. Feature selection results for nonlinear datasets. (a) WDBC dataset. (b) USPS subset, “3” v.s. “5”. (c) USPS subset “6” v.s. “8”.

a one-against-all classifier because the discriminative features will probably be fewer when only two classes are considered.

VI. CONCLUSION

We have presented a new feature selection algorithm for SVMs which works by estimating the stability of a feature’s contribution to some evaluation criterion in an ensemble of SVM classifiers. Unlike the SVM-RFE method, to solve a feature selection problem, the proposed SVM-SE only has to learn a few number of SVM classifiers. To improve the stability of the algorithm, we combine the proposed SVM-SE algorithm with a backward selection procedure. We have also addressed the problem of using a validation dataset to select number of features with optimal prediction power and improve the performance of the proposed algorithms.

Experiments on well studied problems and real-life problems are reported. In all of the reported datasets, the proposed SVM-SE method shows comparable performances compared with the SVM-RFE method. It shows large reduction in computation costs. The combined method shows best results on most of the datasets with a sacrifice of training time.

REFERENCES

[1] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference,* 121–129, 1994.

[2] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence,* 97(1-2):245–271, 1997.

[3] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR,* 3:1157–1182, 2003.

[4] I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: relevancy, filters, and wrappers. In *Ninth International Workshop on Artificial Intelligence and Statistics,* 2003.

[5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning,* 46(1-3):389–422, 2002.

[6] A. Rakotomamonjy. Variable selection using SVM-based criteria. *JMLR,* 3:1357–1370, 2003.

[7] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *15th International Conference on Machine Learning,* 82–90, 1998.

[8] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. *Advances in Neural Information Processing Systems,* 13:668–674, 2000.

[9] O. Bousquet and A. Elisseeff, Stability and generalization. *Journal of Machine Learning Research,* 2:499–526, 2002

[10] L. Breiman. Bagging predictors. *Machine Learning,* 24:123–140, 1996.

[11] R. E. Schapire. Using output codes to boost multiclass learning problems. In *Procedings of the 14th International Conference on Machine Learning,* 313–321, 1997.

[12] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences,* 55(1):119–139, 1997.

[13] L. Breiman, Random forests, Machine learning, vol. 45, no.1, pp. 5–32, 2001. Random forests with ensemble of feature spaces.” Pattern Recognition 47.10 (2014): 3429-3437.

[14] L. Zhang and P. N. Suganthan, Random Forests with ensemble of feature spaces, Pattern Recognition, vol. 47, pp. 3429-3437, 2014.

[15] L. Zhang and P. N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE Transactions on Cybernetics, vol. 45, no. 10, pp. 2165-2176, 2015.

[16] Y. Ren, L. Zhang, and P. N. Suganthan, Ensemble classification and regression recent developments, applications and future directions, IEEE Computational Intelligence Magazine, vol. 11, no. 1, pp. 41-053, Feb. 2016.

[17] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, Do we need hundreds of classifiers to solve real world classification problems, Journal of Machine Learning Research, vol. 15, no. 1, pp. 3133–3181, 2014.

[18] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, 1998.

[19] V. Sindhwani, P. Bhattacharyya, Subrata Rakshit. Information theoretic feature crediting in multiclass support vector machines. In *First SIAM Int. Conf. on Data Mining,* 2001.

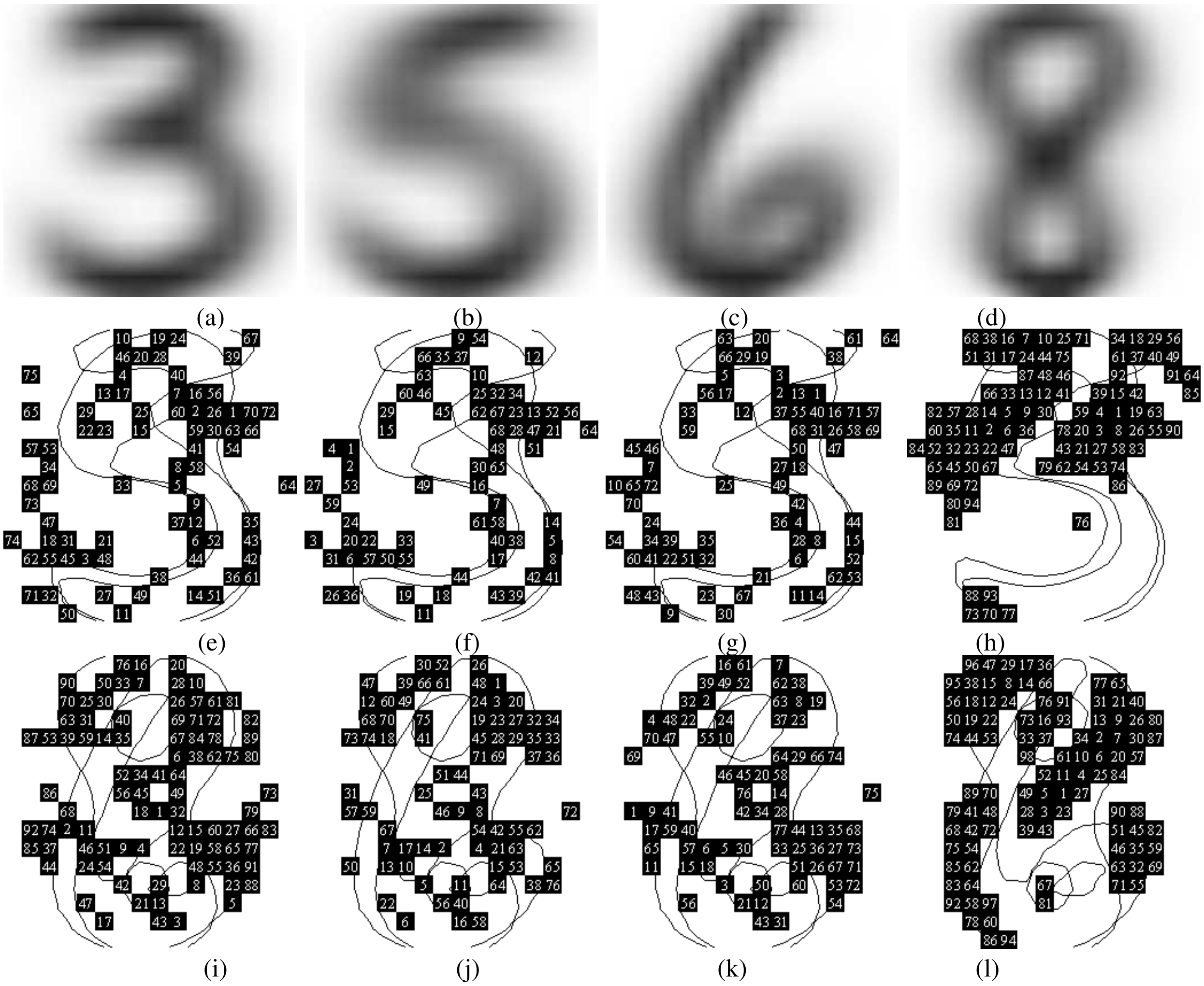


Fig. 3. Feature selection results for USPS subproblems. (a), (b), (c), (d) show the averaged digits, i.e. “3”, “5”, “6”, and “8”, respectively. From (e) to (h) and (i) to (l) are the results of SVM-RFE, SVM-SE, the combined method, and the baseline method. Blackened pixels are the selected features with the ranking order shown on each pixel.

[20] C. Couvreur and Y. Bresler. On the optimality of the backward greedy algorithm for the subset selection problem. *SIAM Journal on Matrix Analysis and Applications,* 21(3):797–808, 2000.

[21] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research,* 43(4):570–577, 1995.

[22] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* The MIT Press, 2002.

[23] C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm, 2001.

[24] T. R. Golub et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science,* 286:531–537, 1999.

[25] N. Sha. *Bolstering CART and Bayesian Variable Selection Methods for Classification.* Ph.D. thesis, Deparment of Statistics, Texas A&M University, 2002.

[26] S. Abe. *Support Vector Machines for Pattern Classification.* Springer-Verlag, 2005.