Journal of Machine Learning Research 3 (2003) 1357-1370 Submitted 05/02; Published 3/03

Variable Selection Using SVM-based Criteria

Alain Rakotomamonjy ALAIN.RAKOTOMAMONJY@INSA-ROUEN.FR
Perception, Systèmes et Information
FRE CNRS 2645
INSA de Rouen
76801 Saint Etienne du Rouvray France

Editors: Isabelle Guyon and André Elisseeff

Abstract

We propose new methods to evaluate variable subset relevance with a view to variable selection. Relevance criteria are derived from Support Vector Machines and are based on weight vector $\|\mathbf{w}\|^2$ or generalization error bounds sensitivity with respect to a variable. Experiments on linear and non-linear toy problems and real-world datasets have been carried out to assess the effectiveness of these criteria. Results show that the criterion based on weight vector derivative achieves good results and performs consistently well over the datasets we used.

Keywords: support vector machines, kernels, variable selection, sensitivity.

1. Introduction

Nowadays, many practical pattern recognition tasks infer knowledge from example data. This knowledge is then used to make predictions about new data or to get a deeper understanding of the system or "concept" that generated the data. Data typically consist of measurements (also referred to as attributes, variables or features) characterizing the system to be modelled. Each example may be represented as a vector in $\mathbb{R}^n$ whose components correspond to such measurements. In a pattern recognition or discrimination problems each example vector is associated with a label specifying the category the example belongs to. Machine learning algorithms estimate dependencies between the examples and their label during a learning process. Progresses made in sensor technology and data management allow researchers to gather data sets of ever increasing sizes, particularly with respect to the number of variables. However, the incremental informative content of such variables is not always significant. This problem may undermine the success of machine learning that is strongly affected by data quality: redundant, noisy or unreliable information may impair the learning process.

The purpose of feature or variable selection is to eliminate irrelevant variables to enhance the generalization performance of a given learning algorithm. The selection of relevant variables may also be useful to gain some insight about the concept to be learned. Other advantages of feature selection include cost reduction of data gathering and storage (in medical applications for instance) and computational speedup.

In this paper we investigate the efficiency of criteria derived from support vector machines (SVMs) for variable selection in application to classification problems. This work can be seen as an extension of the SVM-RFE algorithm (Guyon et al., 2000). Extensive experiments are conducted to compare various methods. The paper is organized as follows: In Section 2 we review SVMs and give details on how variable relevance criteria are derived from the SVM methodology. The associated variable selection algorithm is then presented. Numerical experiments on toy problems and real-world data showing the strength and weakness of different criteria are described in Section 3. Discussions about the questions that have arisen from this work are reported in Section 4.

2. Variable Selection with SVM Criterion

In this section, we explore some possible methods of variable selection using support vector machines. After reviewing the so-called soft margin SVM classifier, we present ranking criteria derived from SVM and an associated algorithm for feature selection. Finally, relationships with other SVM-based feature selection methods are given.

2.1 SVM Classifier

The support vector machine classifier is a binary classifier algorithm that looks for an optimal hyperplane as a decision function in a high-dimensional space (Boser et al., 1992, Vapnik, 1998, Cristianini and Shawe-Taylor, 2000). Consider one has a training data set $\{\mathbf{x}_k, y_k\} \in \mathbb{R}^n \times \{-1, 1\}$ where $\mathbf{x}_k$ are the training examples and y_k the class labels. The method consists in first mapping $\mathbf{x}$ into a high dimensional space via a function Φ, then computing a decision function of the form:

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

by maximizing the distance between the set of points $\Phi(\mathbf{x}_k)$ to the hyperplane parameterized by $(\mathbf{w}, b)$ while being consistent on the training set. The class label of $\mathbf{x}$ is obtained by considering the sign of $f(\mathbf{x})$. For the SVM classifier with misclassified examples being quadratically penalized, this optimization problem can be written as:

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{k=1}^{m} \xi_k^2$$

under the constraint $\forall k, \quad y_k f(\mathbf{x}_k) \geq 1 - \xi_k$. The solution of this problem is obtained using the Lagrangian theory and one can prove that vector $\mathbf{w}$ is of the form:

$$\mathbf{w} = \sum_{k=1}^{m} \alpha_k^* y_k \Phi(\mathbf{x}_k)$$

where α_k^* is the solution of the following quadratic optimization problem:

$$\max_{\alpha} W(\alpha) = \sum_{k=1}^{m} \alpha_k - \frac{1}{2}\sum_{k,\ell}^{m} \alpha_k \alpha_\ell y_k y_\ell \left(K(\mathbf{x}_k, \mathbf{x}_\ell) + \frac{1}{C}\delta_{k,\ell} \right) \tag{4}$$

subject to $\sum_{k=1}^{m} y_k \alpha_k = 0$ and $\forall k, \alpha_k \geq 0$, where $\delta_{k,\ell}$ is the Kronecker symbol and $K(\mathbf{x}_k, \mathbf{x}_\ell) = \langle \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_\ell) \rangle$ is the Gram matrix of the training examples.

The interesting point of SVMs is that they are provided with many statistics that allow to estimate their generalization performance from bounds on the *leave-one-out* error $\mathcal{L}$. The leave-one-out error is the number of classification error produced by the *leave-one-out* procedure which consists in learning a decision function from $m-1$ examples, testing the remaining one and repeating until all elements have served as test example. The *leave-one-out* error is known to be an unbiased estimator of the generalization performance of a classifier trained on $m-1$ examples. One of the most common $\mathcal{L}$ error bounds for SVMs is the radius/margin bound (for decision function with non-zero bias b) (Vapnik, 1998):

$$\mathcal{L} \leq 4R^2\|\mathbf{w}\|^2$$

where R is the radius of the smallest sphere that contains all the mapped data $\Phi(\mathbf{x}_k)$. A tighter bound named "span estimate" is also available and is based on the distance S_p between a mapped support vector $\Phi(\mathbf{x}_p)$ and the span of all other support vectors (Vapnik and Chapelle, 2000). The following equation holds:

$$\mathcal{L} \leq \sum_p \alpha_p^* S_p^2$$

where S_p^2, for SVM with quadratic slack variables ξ, is related to the extended matrix of the dot product between support vectors

$$\widetilde{K}_{SV} = \begin{pmatrix} \mathbf{K} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{pmatrix}$$

by the equation $S_p^2 = 1/(\widetilde{K}_{SV}^{-1})_{pp}$

2.2 SVM-RFE Algorithm

The SVM-RFE algorithm has been recently proposed by Guyon et al. (2000) for selecting genes that are relevant for a cancer classification problem. The goal is to find a subset of size r among d variables ($r < d$) which maximizes the performance of the predictor. The method is based on a backward sequential selection. One starts with all the features and removes one feature at a time (in their paper, due to the large amount of genes, they remove chunks of features) until r features are left. The removed variable is the one whose removal minimizes the variation of $\|\mathbf{w}\|^2$. Hence, the ranking criterion R_c for a given variable i is:

$$\left|\|\mathbf{w}\|^2 - \|\mathbf{w}^{(i)}\|^2\right| = \frac{1}{2}\left|\sum_{k,j}\alpha_k^*\alpha_j^* y_k y_j K(\mathbf{x}_k,\mathbf{x}_j) - \sum_{k,j}\alpha_k^{*(i)}\alpha_j^{*(i)} y_k y_j K^{(i)}(\mathbf{x}_k,\mathbf{x}_j)\right| \tag{8}$$

where $K^{(i)}$ is the Gram matrix of the training data when variable i is removed ($K_{k,j}^{(i)} = \langle\Phi(\mathbf{x}_k^{(i)}),\Phi(\mathbf{x}_j^{(i)})\rangle$) and $\alpha_k^{*(i)}$ is the corresponding solution of Equation (4). For the sake of simplicity and to reduce computational complexity of this algorithm, the $\alpha_k^{*(i)}$ is supposed to be equal to α_k^* even if a variable has been removed. The authors also stated that in order for RFE to work variable scaling is needed. From Equation (8), one can consider that the removed variable is the one which has the least influence on the weight vector norm. Hence, this method is similar to those employed in neural networks in the sense that the ranking criterion is the sensitivity of $\|\mathbf{w}\|^2$ with respect to a variable.

2.3 Algorithm for Variable Ranking with SVM

Variable selection algorithms require a ranking criterion to rank variables. In many papers, bounds on the $\mathcal{L}$ error have been used for model selection (Duan et al., 2002) and recently Weston et al. (2001b) used the radius/margin bound for feature selection using a gradient descent algorithm. This idea can therefore be extended to other bounds of the generalization error. In this paper, we will investigate three criteria C_t which are either the weight vector$\|\mathbf{w}\|^2$, the radius/margin bound $R^2\|\mathbf{w}\|^2$ or the span estimate. These criteria give either an estimation of the generalization performance (the bounds) or an estimation of the dataset separability. Furthermore, similarly to neural-networks based variable selection (Leray and Gallinari, 1999), two approaches can be proposed for each criterion:

- Zero-order method: in this case, the criterion C_t is directly used for variable ranking, and the methods consists in identifying the variable that produces the smallest value of C_t when removed. The ranking criterion then becomes $R_c(i) = C_t^{(i)}$ with $C_t^{(i)}$ being the criterion value when variable i has been removed.

- First-order method: one uses the derivatives of the criterion C_t with regards to a variable. In other words, this approach differs from the previous one since a variable is ranked according to its influence on the criterion which is measured with the absolute value of the derivative. In this case, the ranking criterion is $R_c(i) = |\nabla C_t|$.

The zero-order criteria based on bounds have already been used for feature selection associated with different search space algorithm (Weston et al., 2001b) whereas the first-order ones are rather new for the purpose of feature selection.

Similarly to SVM-RFE, the problem of searching the "best" r variables is solved by means of a greedy algorithm based on backward selection (Kohavi and John, 1997). 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 (Couvreur and Bresler, 2000). Hence, the algorithm starts with all features and repeatedly removes a feature until r features are left or all variables have been ranked (see Figure 1). In the zero-order method, one suppresses the feature whose removal minimizes the criterion whereas in first-order methods, one removes the variable to which the criterion is less sensitive. For instance, in the zero-order $\|\mathbf{w}\|^2$ case, the ranking term is:

$$R_c(i) = \|\mathbf{w}^{(i)}\|^2 = \sum_{k,j} \alpha_k^{*(i)} \alpha_j^{*(i)} y_k y_j K^{(i)}(\mathbf{x}_k, \mathbf{x}_j) \tag{9}$$

where $K^{(i)}$ is again the Gram matrix of the training data when the variable i has been removed. Note that in this case, the criterion should be evaluated with the appropriate $\alpha_k^{*(i)}$. Similarly to SVM-RFE and to reduce time complexity we consider that these parameters are equal to α_k^* during the evaluation of $R_c(i)$. However, it would still be interesting to consider how the SVM retraining at each subset evaluation affects the results and some experiments using the true $\alpha_k^{*(i)}$ will be carried out.

In the first-order case, the ranking term for $\|\mathbf{w}\|^2$ case criterion would be:

$$R_c(i) = \left|\nabla\|\mathbf{w}\|^2\right|$$

1. Initialization: $Ranked = []$; $Var = [1,\ldots,N]$
2. **repeat**
   (a) Train a SVM classifier with all the training data and the variables *Var*
   (b) **for** all variables in *Var*, **do** evaluate the ranking criterion $R_c(i)$ of variable i **endfor**
   (c) $best = \arg\min_i R_c$
   (d) rank the variable that minimizes R_c: $Ranked = [best\, Ranked]$;
   (e) remove the variable that minimizes R_c from the selected variables set: $Var = [1,\ldots,best-1,best+1,\ldots,N]$
3. **until** *Var* is not empty

Figure 1: Outline of the SVM-based feature selection algorithm.

2.4 Calculating the Gradient with Regards to a Scaling Factor ν

For the first-order criterion, our aim is to measure the sensitivity of a given criterion with respect to a variable. A possible approach is to introduce a virtual scaling factor and to compute the gradient of a criterion with respect to that scaling factor ν. The latter acts as a componentwise multiplicative term (whose value is 1) on the input variables and thus $k(\mathbf{x},\mathbf{x}')$ becomes:

$$k(\nu\cdot\mathbf{x},\nu\cdot\mathbf{x}')$$

where $\cdot$ denotes the componentwise vector product. Consequently, one obtains the following derivatives for a Gaussian Kernel $k(\nu\cdot\mathbf{x},\nu\cdot\mathbf{x}') = \mathbf{e}^{-\frac{\|\nu\cdot\mathbf{x}-\nu\cdot\mathbf{x}'\|^2}{2\sigma^2}}$:

$$\frac{\partial k}{\partial \nu_i} = -\frac{1}{\sigma^2}(\nu_i x_i - \nu_i x_i^{'})^2 k(\mathbf{x},\mathbf{x}') = -\frac{1}{\sigma^2}(x_i - x_i^{'})^2 k(\mathbf{x},\mathbf{x}')$$

where we used the fact that $\nu_i = 1$. Then, one needs to evaluate the gradient of the bounds with regards to a variable ν_i and for a given criterion C_t the ranking term becomes :

$$R_c(i) = \left|\frac{\partial C_t(\alpha,b)}{\partial \nu_i}\right| \tag{11}$$

where C_t is either $\|\mathbf{w}\|^2$, $R^2\mathbf{w}^2$ or $\sum_p \alpha_p^* S_p^2$ and depends on the solution of Equation (4) and the bias b. Details of the derivatives computation for a given criterion are presented in the report of Rakotomamonjy (2002), and they have been obtained using the results of Bengio (2000) and Chapelle et al. (2002). Here, we only give the final results:

- weight vector gradient:

$$R_c(i) = \left|\sum_{k,j} \alpha_k^* \alpha_j^* y_k y_j \frac{\partial k(\nu\cdot\mathbf{x_k},\nu\cdot\mathbf{x_j})}{\partial \nu_i}\right|$$

- radius/margin gradient:

$$R_c(i) = \left| \|\mathbf{w}\|^2 \sum_{k,j} (\beta_k \beta_j - \beta_k \delta_{k,j}) \frac{\partial k(\nu \cdot \mathbf{x_k}, \nu \cdot \mathbf{x_j})}{\partial \nu_i} + R^2 \sum_{k,j} \alpha_k^* \alpha_j^* y_k y_j \frac{\partial k(\nu \cdot \mathbf{x_k}, \nu \cdot \mathbf{x_j})}{\partial \nu_i} \right|$$

where R^2 is the optimal objective function of the following problem:

$$\begin{array}{ll} \max_\beta & \sum_k \beta_k k(\nu \cdot \mathbf{x}_k, \nu \cdot \mathbf{x}_k) - \sum_{k,j} \beta_k \beta_j k(\nu \cdot \mathbf{x_k}, \nu \cdot \mathbf{x_j}) \\ \text{s.t} & \sum_k \beta_k \quad \text{and} \quad \beta_k \geq 0 \; \forall k \end{array}$$

- span estimate gradient:

$$R_c(i) = \left| \sum_{p=1}^{\ell} 2 \left(-H^{-1} \frac{\partial H}{\partial \nu_i} \alpha^* \right)_{pp} S_p^2 + \alpha_p^* S_p^4 \left(\widetilde{K}_{SV}^{-1} \frac{\partial \widetilde{K}_{SV}}{\partial \nu_i} \widetilde{K}_{SV}^{-1} \right)_{pp} \right|$$

where H is the following matrix $H = \begin{pmatrix} \mathbf{K}^Y & \mathbf{Y} \\ \mathbf{Y}^T & 0 \end{pmatrix}$ and $\mathbf{K}_{kj}^Y = y_k y_j k(\nu \cdot \mathbf{x_k}, \nu \cdot \mathbf{x_j})$

As noticed previously all these gradients are computed for $\nu = (1, .., 1)$. In what follows, we use the notation ∇C_t to denote these first order criteria where C_t is either $\|\mathbf{w}\|^2$, $R^2\mathbf{w}^2$ or $\sum_p \alpha_p^* S_p^2$.

2.5 Relation to Other SVM-Based Feature Selection Methods

In addition to SVM-RFE, several algorithms for feature selection based on SVM are already available. For instance, Weston et al. (2001b) propose a method based on finding the best variable subset which minimizes the $R^2\mathbf{w}^2$ bound. For this criterion, their method differs from ours in the variable space search algorithm. In fact instead of using a greedy algorithm, they use a gradient descent to minimize the bound with respect to a scaling vector associated to variables.

In the linear case, an interesting relation links SVM-RFE and our method when using the derivatives of $\|\mathbf{w}\|^2$ with respect to a virtual scaling factor. The RFE criterion for a variable i is $R_c(i) = w_i^2$ whereas the gradient of $\|\mathbf{w}\|^2$ with respect to ν_i gives $R_c(i) = |-w_i^2|$ (ν_i being the scaling factor associated to variable i). Thus, SVM-RFE and gradient of $\|\mathbf{w}\|^2$ are identical as they have the same ranking criterion.

In addition, one should note that SVM-RFE and the zero-order $\|\mathbf{w}\|^2$ criterion are identical since the first sum in Equation (8) is constant during the evaluation of $R_c(i)$. For this reason, results concerning SVM-RFE are not reported in the experimental section.

3. Numerical Experiments

The experiments that we report here use artificial and real-world datasets. We have compared the classification performance of the different ranking criteria for feature selection associated to a SVM classifier with quadratic slack variables ξ_i as a predictor. In addition, in all experiments the results of a stand-alone SVM classifier are presented along with another SVM associated with a method for feature selection based on correlation coefficients.

	Training set size				
Methods	10	20	30	40	50
SVM	36.58%±2%	30.89%±2%	25.46%±2%	22.22%±2 %	19.40%±2%
Corr	32.33%±13%	17.00%±7%	14.30%±4%	14.69%±2 %	14.76%±2%
$\|\|\mathbf{w}\|\|^2$	35.63%±15%	14.79%±13%	5.99%±5%	**4.40%±3%**	4.19%±3%
$R^2\mathbf{w}^2$	32.83%±15%	13.60%±12%	**5.82%±5%**	4.53%±3%	**4.04%±2%**
S_p^2 Est.	38.92%±13%	21.14%±15%	14.22%±12%	10.68%±9%	7.34%±6%
$\|\|\mathbf{w}_i\|\|^2$	32.39%±15%	**13.05%±11%**	7.14%±6%	6.30%±5%	5.11%±4%
$R_i^2\mathbf{w}_i^2$	**31.21%±15%**	19.13%±12%	14.55%±9%	14.10%±9%	13.13%±9%
S_{pi}^2 Est.	50.02%±0.5%	50.02%±0.5%	49.44%±2%	49.83%±2%	49.49%±2%
$\nabla\|\|\mathbf{w}\|\|^2$	32.39%±15%	**13.05%±11%**	7.14%±6%	6.30%±5%	5.11%±4%
$\nabla R^2\mathbf{w}^2$	33.50%±15%	36.87%±16%	43.69%±17%	46.28%±10%	46.81%±9%
$\nabla S_p^2 Est$	41.51%±12%	23.85%±13%	15.91%±10%	13.76%±9%	13.16%±7%

Table 1: Mean and standard deviation of test error for feature selection on a synthetic linear problem using different criteria and different training set sizes. The methods are: (a) SVM: standard SVM,(b) Corr: SVM with correlation coefficients feature selection algorithm, (c) $\|\mathbf{w}\|^2$, $R^2\mathbf{w}^2$ and S_p^2 Est zero-order criterion with retraining, (d) $\|\mathbf{w}_i\|^2$, $R_i^2\mathbf{w}_i^2$ and S_{pi}^2 Est zero-order criterion. (e) $\nabla\|\mathbf{w}\|^2$, $\nabla R^2\mathbf{w}^2$ and ∇S_p^2 Est first-order criterion.

3.1 Toy Experiments

For toy experiments, we used the datasets described in the work of Weston et al. (2001a,b), which allows comparing the results obtained with our criteria to those described in these references. A precise description of these synthetic data can be found in Weston et al. (2001b). In the 2-class linear problem, the input data are composed of 202 variables from which only 6 are relevant whereas, in the nonlinear one, 52 variables are available and only the first two are relevant. In both cases, 10000 points have been generated. Only a randomly-chosen small proportion of them are used as a training set and the rest are included in a test set. The training set has been normalized to get zero mean and unit standard deviation. The test set is normalized according to the training set normalization parameters.

For both feature selection and classification, we used a linear SVM for the linear problem and a Gaussian kernel with $\sigma = 3$ for the nonlinear problem. In both linear and non-linear cases, the hyperparameter C has been set sufficiently high (respectively $C = 100000$ and $C = 1000$) in order to keep training error low. After feature selection has been performed, only the two top-ranked variables are provided to the predictor.

Table 1 and Table 2 present the mean and the standard deviation of the test error over 100 trials for each training set size. For both datasets, SVM without feature selection overfits. When considering the baseline feature selection method based on correlation coefficients, the test error becomes significantly lower in the linear case but does not decrease in the nonlinear problem. This is simply due to the incapability of this feature selection method to represent variable correlation in a nonlinear context.

Retraining a SVM at each step should increase the capacity of a zero-order criterion to select the relevant features because the true α^* are used (e.g. see Equation 9) . This is clear in the linear case particularly when the number of training points increases but it is not so obvious in the nonlinear case. In fact for $\|\mathbf{w}\|^2$ and $R^2\mathbf{w}^2$ criterion, the test errors are always higher when

	Training set size					
Methods	10	20	30	40	50	100
SVM	49.20%±1%	48.39%±1%	47.84%±1%	47.52%±1 %	46.81%±1%	45.54%±1 %
Corr	49.43%±3%	49.45%±4%	49.07%±5%	48.80%±6 %	48.78%±6%	49.48%±5 %
$\|\|\mathbf{w}\|\|^2$	49.24%±3%	43.12%±13%	33.31%±18%	20.82%±19%	9.56%±10%	**4.21%±0.7%**
$R^2\mathbf{w}^2$	49.22%±3%	42.78%±14%	32.46%±19%	20.39%±19%	8.71%±10%	$*$
S_p^2 Est.	48.82%±4%	44.08%±12%	33.15%±17%	20.86%±17%	11.26%±12%	$*$
$\|\|\mathbf{w}_i\|\|^2$	48.29%±5%	37.18%±15%	24.42%±19%	17.75%±18%	8.79%±12%	6.08%±8%
$R_i^2\mathbf{w}_i^2$	48.58%±4%	34.59%±18%	21.50%±19%	13.64%±16%	**8.61%±11%**	**4.21%±0.7%**
S_{pi}^2 Est.	49.86%±2%	49.96%±0.5%	49.21%±4 %	48.67%±6 %	49.37%±4%	47.07%±10%
$\nabla\|\|\mathbf{w}\|\|^2$	48.62%±5%	**32.62%±17%**	**18.57%±16%**	**12.53%±13%**	9.43%±12%	9.41%±12%
$\nabla R^2\mathbf{w}^2$	**48.22%±6%**	33.67%±17%	19.15%±17%	13.61%±15%	11.62%±14%	17.75%±19%
$\nabla S_p^2 Est$	48.92%±5%	43.65%±12%	44.91%±12%	44.42%±12%	46.12%±15%	50.00%±1%

Table 2: Mean and standard deviation of test error for feature selection on a synthetic non linear problem using different criteria and different training set sizes. The methods are: (a) SVM: standard SVM,(b) Corr: SVM with correlation coefficients feature selection algorithm, (c) $\|\mathbf{w}\|^2$, $R^2\mathbf{w}^2$ and $S_p^2 Est$ zero-order criterion with retraining, (d) $\|\mathbf{w}_i\|^2$, $R_i^2\mathbf{w}_i^2$ and $S_{pi}^2 Est$ zero-order criterion, (e) $\nabla\|\mathbf{w}\|^2$, $\nabla R^2\mathbf{w}^2$ and $\nabla S_p^2 Est$ first-order criterion. An asterisk $*$ indicates that full experiments had not been carried out because of excessive time.

retraining is performed. The span estimate criterion does well only with retraining. This is merely explained by the tight relation of this span estimate with the value of α^* (Vapnik and Chapelle, 2000) and thus keeping α^* fixed during the evaluation of $R_c(i)$ leads to a wrong estimation of variable relevance.

Without retraining, in the linear case the $\nabla\|\mathbf{w}\|^2$ (which is identical to the $\|\mathbf{w}\|^2$ criterion) outperforms other methods. For the nonlinear problem, $\nabla\|\mathbf{w}\|^2$ and $R^2\mathbf{w}^2$ criteria share the best performance depending on the size of the training set.

3.2 Real-World Data

In order to assess the effectiveness of the proposed criteria, experiments on real-world datasets have also been performed.

3.2.1 BENCHMARK DATASETS

At first, we compared performances on some of the real-world benchmark datasets used by Rätsch et al. (2001). The methodology we followed consisted for each realization of the datasets in: (1) performing a variable ranking and (2) measuring the test error of an SVM classifier when this predictor is provided with an increasing number of ranked variables. The hyperparameters of the SVM have been set to the values found by Rätsch et al. (2001) with their cross-validation procedure. A mean test error is then obtained by averaging the results over the 100 realizations. Figures 2 and 3 represent this mean test error for an increasing number of ranked features used for learning. They show that for these problems, one can achieve better or similar performance using fewer variables and that the span estimate criterion gives poor results and seems not to be able to rank variables appropriately without retraining. Error bars of standard deviation have not

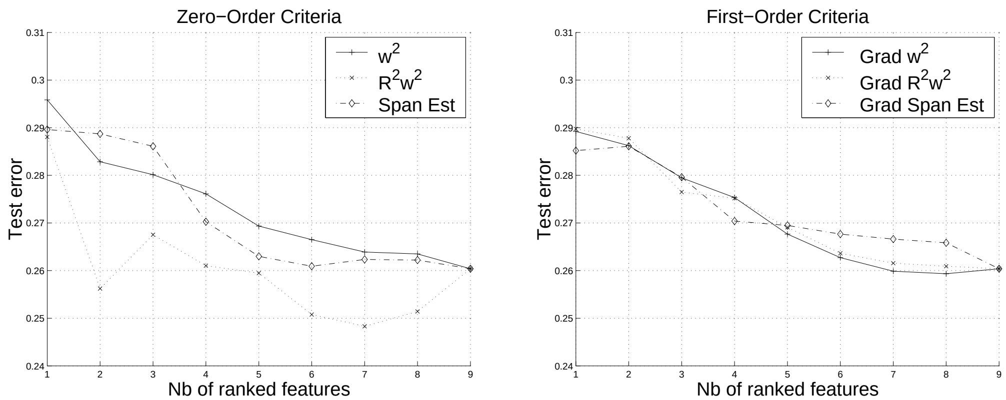

Figure 2: Mean of test error for a feature selection problem on a real-world problem. Mean test errors for Breast Cancer Data *vs.* the number of ranked variables used for training (C=15, $\sigma = 5$). (left) $\|\mathbf{w}_i\|^2$, $R_i^2\mathbf{w}_i^2$ and S_{pi}^2Est zero-order criterion. (right) $\nabla\|\mathbf{w}\|^2$, $\nabla R^2\mathbf{w}^2$ and $\nabla S_p^2 Est$.

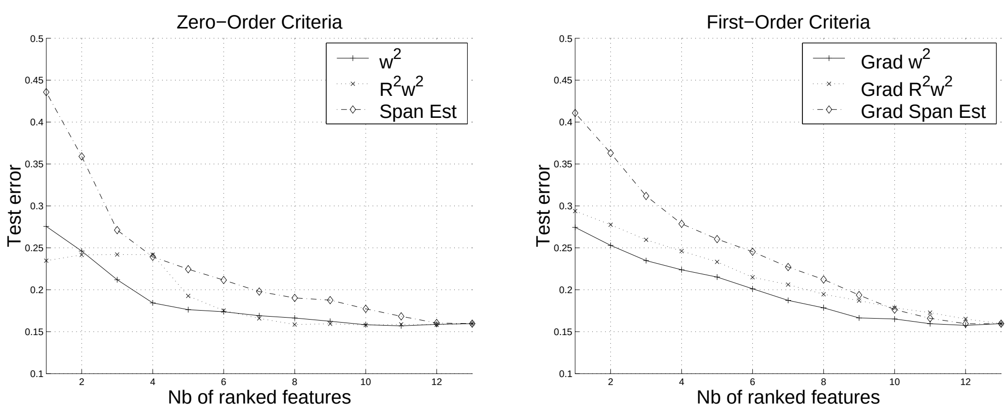

Figure 3: Mean of test error of feature selection on real-world problem. Mean test error for Heart Data *vs.* to the number of ranked variables used for training (C=3.16, $\sigma = 7.7$) (left) $\|\mathbf{w}_i\|^2$, $R_i^2\mathbf{w}_i^2$ and $S_{pi}^2 Est$ zero-order criterion. (right) $\nabla\|\mathbf{w}\|^2$, $\nabla R^2\mathbf{w}^2$ and $\nabla S_p^2 Est$.

been plotted for the sake of clarity. However, we can state that for the Breast Cancer dataset the standard deviation is rather stable with regards to the number of features used for classification (around 5%) whereas for the Heart dataset it tends to decrease from 7% to 4%.

	Number of variables					
Methods	20	50	100	250	500	1000
Corr	21.58%±11%	22.08%±10%	20.83%±11%	17.83%±9 %	18.42%±9%	16.75%±9%
$R_i^2\mathbf{w}_i^2$	19.67%±11%	17.33%±9%	16.17%±9%	16.66%±9%	16.53%±8%	**15.91%±8%**
S_{pi}^2 Est.	22.66%±12%	34.25%±14%	30.50%±12%	23.83%±10%	18.91%±9%	17.67%±9%
$\nabla\mathbf{w}^2$	**17.67%±9%**	**15.66%±9%**	**15.17%±10%**	16.50%±9%	**16.08%±9%**	16.25%±9%
$\nabla R^2\mathbf{w}^2$	20.33%±13%	17.83%±10%	16.41%±9%	**15.58%±9%**	16.16%±9%	16.16%±8%
$\nabla S_p^2 Est$	20.50%±11%	17.00%±10%	16.33%±9%	16.75%±8%	16.66%±8%	16.41%±9%

Table 3: Mean and standard deviation of test error for a feature selection problem on a microarray Colon Cancer Dataset. The methods are: (a) Corr: SVM with correlation coefficients feature selection algorithm, (b) $R_i^2\mathbf{w}_i^2$ and $S_{pi}^2 Est$ zero-order criterion, (c) $\nabla\|\mathbf{w}\|^2$, $\nabla R^2\mathbf{w}^2$ and $\nabla S_p^2 Est$ first-order criterion.

3.2.2 MICROARRAY DATA

Experiments on DNA microarray analysis have also been performed. The data we used concerned two classification problems, the first one dealing with normal and cancerous colon tissue and the second one with a lymphoma problem. These datasets have already been used for benchmarking feature selection algorithms (for example, see Weston et al., 2001a).

The colon cancer tissue problem is composed of 62 observations (22 normal and 40 cancerous) described by 2046 features. Following the step of Weston et al. (2001a), the training set and the test set are obtained by splitting the dataset into two groups of respectively 50 and 22 elements, while ensuring that the proportions of positive and negative classes are similar in both sets. 100 trials are carried out with random splitting of dataset. In order to speed up the feature selection procedure, half of the variables are removed at each step until 100 variables remain still to be ranked. Then variables are removed one at a time. The predictor is a linear SVM (with $C = 10^6$) and it achieves an average test error of $16.4\% \pm 8\%$. Results with an increasing number of features provided to the predictor are described in Table 3. The performances are in the same range but one can see that the criterion $\nabla\|\mathbf{w}\|^2$ slightly outperforms the others. Again, retraining does not improve all that much the ability of ranking relevant variables for any of the zero-order criteria.

The lymphoma problem is based on 4026 variables describing 96 observations (62 and 34 of which are respectively considered as abnormal and normal). The data is split into two sets of sizes 60 and 36 with similar proportions of abnormal and normal examples. The same methodology as in the colon cancer problem is followed and a linear SVM (with $C = 10^6$) gives a test error of $7.25\% \pm 4.1\%$. Results obtained with a different number of features and criteria for feature selection are given in Table 4. It seems that the $\nabla S_p^2\ Est$ criterion performs better than other criteria and achieves the best performance with 5.58% test error with only 250 variables. However, it should be noted that with the $\nabla\|\mathbf{w}\|^2$ criterion good performance can be achieved using only 20 variables.

4. Discussions

So far, we have presented different criteria for feature selection and compared them experimentally. In this section, we discuss some points that naturally arise from this work.

	Number of variables					
Methods	20	50	100	250	1000	2000
Corr	21.58%±11%	13.30%±5.7%	9.11%±5.0%	7.50%±4.8%	6.89%±4.6%	6.92%±4.4 %
$R_i^2\mathbf{w}_i^2$	8.83%±4.4%	6.86%±4.3%	6.44%±4.1%	6.33%±4%	**6.58%±4.1%**	6.94%±4.2%
S_{pi}^2Est	28.58%±8.3%	27.25%±8.2%	19.94%±7.4%	11.47%±5.4 %	6.94%±3.9%	**6.39%±4.2%**
$\nabla\mathbf{w}^2$	**7.72%±4.0%**	**6.58%±4.6%**	**6.11%±4.6%**	6.67%±4.2%	6.97%±4.2%	7.22%±4.2%
$\nabla R^2\mathbf{w}^2$	12.25%±5.4%	8.08%±5.0%	6.36%±4.4%	6.16%±4.2%	6.91%±4.1%	7.14%±4.2%
∇S_p^2Est	12.05%±5.5%	7.64%±4.6%	6.13%±4.0%	**5.58%±4.1%**	6.63%±4.0%	7.00%±4.3%

Table 4: Mean and standard deviation of test error for a feature selection problem on a microarray Lymphoma MicroArray Dataset. The methods are: (a) Corr: SVM with correlation coefficients feature selection algorithm, (b) $R_i^2\mathbf{w}_i^2$ and S_{pi}^2Est zero-order criterion, (c) $\nabla\|\mathbf{w}\|^2$, $\nabla R^2\mathbf{w}^2$ and ∇S_p^2Est first-order criterion with respect to a scaling factor.

4.1 How Many Ranked Features Must be Used?

Up to now, the question of how many ranked features must be provided to the predictor has not been addressed. Our aim is not to investigate this point completely but rather to suggest some possible solutions.

- The most straightforward idea is to use a *leave-one-out* procedure or a validation set to estimate the generalization error with regards to the number of features and choose the number of variables which minimizes the test error. However, this method is computationally expensive.

- Another approach is to use one of the SVM upper bound of the $\mathcal{L}$ error (for instance, $R^2\mathbf{w}^2$) for selecting the best model. The drawback is that these bounds are usually loose bounds and they do not always reflect the generalization performance behavior.

- A classical approach already described in feature selection literature for backward elimination is to stop removing variables when the ranking term increases significantly as a variable is removed. Typically, one measures the ranking term $R_c(i)$ and keeps on eliminating variables as long as $R_c(i)$ is below a threshold. For instance, this means that when using first-order criterion one can keep on removing variables as long as the derivative norm is below a given threshold.

Figure 4 shows how these stopping techniques work on the toy nonlinear dataset problem. In this problem there are only two relevant variables. As expected, the validation error is the best method and because of their looseness, upper bounds of the $\mathcal{L}$ error perform poorly. These findings are similar to those obtained for model selection or hyperparameters tuning (Duan et al., 2002). Figure 4 (right) plots the criterion value when a given feature has been removed with respect to the number of features that still have to be ranked (note that we use a logarithmic scale and thus values below 0 correspond to very small ranking term values). Supposing that the chosen threshold is 100, this method would have kept around 10 variables regardless of the criterion used. However another hyperparameter (the threshold) must be tuned, which limits the advantage of the method.

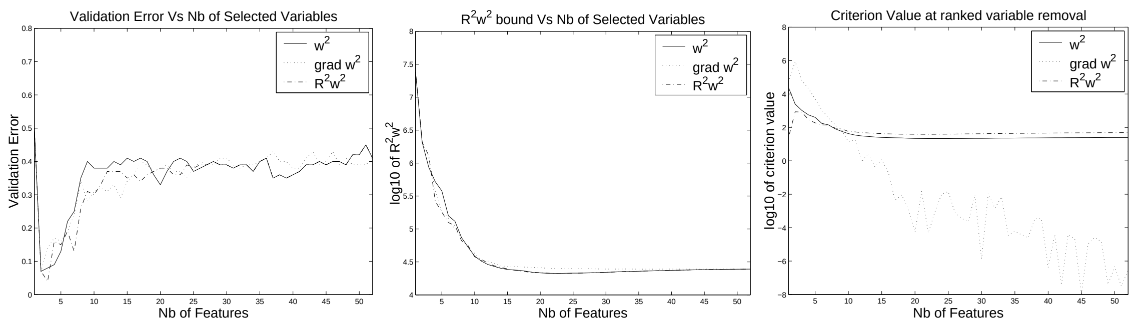

Figure 4: Different ways of choosing the number of ranked features to be provided to the predictor. Results for $\|\mathbf{w}\|^2$, $\nabla\|\mathbf{w}\|^2$ and $R^2\mathbf{w}^2$ are depicted. (left) Validation error. (middle) $\mathcal{L}$ error Estimation with $\frac{1}{m}R^2\mathbf{w}^2$. (right) Criterion value when variable has been removed.

4.2 Influence of SVM Hyperparameters

SVMs involve several hyperparameters (e.g. Gaussian kernel parameter σ, degree d of a polynomial kernel, slack variables penalization C) that have to be tuned to achieve the best generalization performance. This is a crucial issue that is usually solved by minimizing a validation error, a leave-one-out error or an upper bound on the generalization error (Duan et al., 2002, Chapelle et al., 2002, Bengio, 2000). In our feature selection algorithm, these hyperparameters play an important role as they are related to a criterion value through Equation (4). An example of the influence of these hyperparameters on the test error is depicted in Figure 5. The plots represent the mean test error of the nonlinear toy problem using 3 different criteria. The settings are the same as in the experiment involving this data but only C or σ is varying over a range of values. These figures clearly show that the problem of model selection is a crucial issue that must be addressed accurately. This point is beyond the scope of this paper but the most intuitive way of solving this problem is by minimizing a validation error. However, this model selection phase can be computationally very expensive since it involves the SVM hyperparameters as well as the choice of the number of features to be used as stated in the previous section.

5. Conclusion

This paper has presented different criteria for variable selection algorithms. These criteria are derived from generalization error bounds of the SVM theory: weight vector norm $\|w\|^2$ and upper bounds of the *leave-one-out* error. Drawing inspiration from the neural networks community, we have derived zero-order and first-order criteria. The former employ directly the original bounds as criteria whereas the latter employ a derivative of zero-order criteria. Hence, in some sense, first-order methods measure the sensitivity of a zero-order criterion to a variable. The heuristics underlying our selection methods is that “variables relevant to the concept should affect generalization error bounds more than irrelevant ones.” We have investigated the performance of the proposed criteria through experimental comparisons. The main conclusions are:

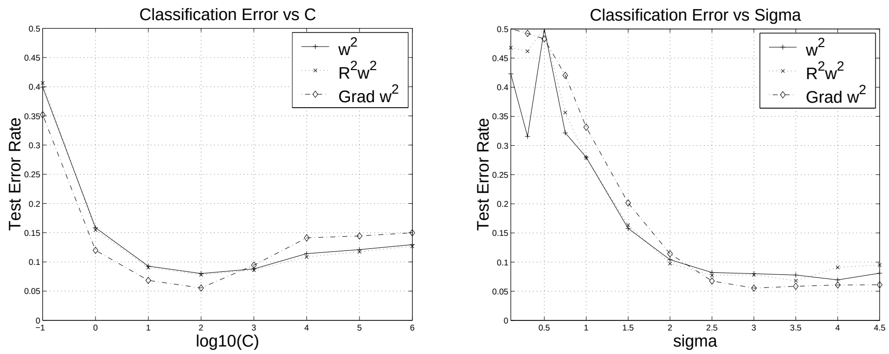

Figure 5: Plots of the influence of C and σ on the test error (averaged over 100 realizations) using the nonlinear toy dataset (training set size is 50). (left) Feature selection and classification error with $\sigma = 3$ vs C. (right) Feature selection and classification error with $C = 100$ vs σ.

- The $\nabla\|\mathbf{w}\|^2$ criterion performs consistently well over all the datasets we used. In addition, it implements a criterion similar to the SVM-RFE criterion in the sense that SVM-RFE measures the sensitivity of $\|\mathbf{w}\|^2$ to a variable by computing the change in $\|\mathbf{w}\|^2$ when this given variable has been removed. In the linear case, these two methods become identical. Lastly, as it has the lowest time complexity (Rakotomamonjy, 2002), it may be the most useful one for practical applications.

- When a large number of training examples is available, retraining significantly improves the ability of zero-order criterion to select relevant variables at the expense of increased time complexity. Surprisingly, retraining does not always improve the ability of SVMs to select these relevant variables regardless of the criterion used. In cases in which the training set size is small, using the exact $\alpha^{*(i)}$ in the processing of the ranking term $R_c(i)$ tends to decrease the performance. Intuitively, one may justify this behavior by the overfitting effects occurring due to the small number of data and the large number of variables. However, this point is far from being clear and some further analysis is needed in order to fully understand this issue.

Examples on real-world data demonstrate the usefulness of the proposed criteria. The performance obtained without variable selection is either closely matched or improved using far fewer variables selected with the proposed algorithms.

Our algorithms rely on a backward feature selection, which is computationally tractable but not necessarily optimal. We may improve the performance of our algorithms by using an alternate search strategy or by combining the feature selection process and the learning process into an overall optimization problem. More work should also be devoted to the problem of hyperparameter selection in conjunction to that of feature selection. Finally, further investigations should focus on the theoretical analysis of the algorithm, as well as making comparisons with other methods.

Acknowledgements

The author would like to thank the referees and the editors of this issue for their comments and suggestions, and Jason Weston for making the datasets he used publicly available.

References

Y. Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12:1889–1900, 2000.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukerjhee. Choosing multiple parameters for svm. *Machine Learning*, 46(1-3):131–159, 2002.

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.

N. Cristianini and J. Shawe-Taylor. *Introduction to Support Vector Machines*. Cambridge Univeristy Press, 2000.

K. Duan, S.S. Keerthi, and A.N. Poo. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, To appear, 2002.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 2000.

R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, (97):273–324, 1997.

P. Leray and P. Gallinari. Feature selection with neural networks. *Behaviormetrika*, 26(1):145–166, 1999.

A. Rakotomamonjy. Variable selection using svm based criteria. Technical Report 02-004, Insa de Rouen Perception Système Informations, http://asi.insa-rouen.fr/˜arakotom, 2002.

G. Rätsch, T. Onoda, and K-R Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9), 2000.

J. Weston, A. Elisseeff, and B. Scholkopf. Use of the ℓ_0-norm with linear models and kernel methods. Technical report, BIOwulf Technical Report, 2001a.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *Advances in Neural Information Processing Systems 13*, 2001b.