Mach Learn (2008) 70: 1–20
DOI 10.1007/s10994-007-5025-7

Feature selection via sensitivity analysis of SVM probabilistic outputs

Kai-Quan Shen · Chong-Jin Ong · Xiao-Ping Li · Einar P.V. Wilder-Smith

Received: 24 May 2006 / Revised: 6 July 2007 / Accepted: 9 September 2007 /
Published online: 3 October 2007

Abstract Feature selection is an important aspect of solving data-mining and machine-learning problems. This paper proposes a feature-selection method for the Support Vector Machine (SVM) learning. Like most feature-selection methods, the proposed method ranks all features in decreasing order of importance so that more relevant features can be identified. It uses a novel criterion based on the probabilistic outputs of SVM. This criterion, termed Feature-based Sensitivity of Posterior Probabilities (FSPP), evaluates the importance of a specific feature by computing the aggregate value, over the feature space, of the absolute difference of the probabilistic outputs of SVM with and without the feature. The exact form of this criterion is not easily computable and approximation is needed. Four approximations, FSPP1-FSPP4, are proposed for this purpose. The first two approximations evaluate the criterion by randomly permuting the values of the feature among samples of the training data. They differ in their choices of the mapping function from standard SVM output to its probabilistic output: FSPP1 uses a simple threshold function while FSPP2 uses a sigmoid function. The second two directly approximate the criterion but differ in the smoothness assumptions of criterion with respect to the features. The performance of these approximations, used in an overall feature-selection scheme, is then evaluated on various artificial

Editor: Risto Miikkulainen.

K.-Q. Shen · C.-J. Ong (✉) · X.-P. Li
BLK EA, #07-08, Department of Mechanical Engineering, National University of Singapore,
9 Engineering Drive 1, Singapore 117576, Singapore
e-mail: mpeongcj@nus.edu.sg

K.-Q. Shen
e-mail: shen@nus.edu.sg

X.-P. Li
e-mail: mpelixp@nus.edu.sg

E.P.V. Wilder-Smith
Neurology, National University Hospital, 5 lower Kent Ridge Road, Singapore 119074, Singapore
e-mail: mdcwse@nus.edu.sg

problems and real-world problems, including datasets from the recent Neural Information Processing Systems (NIPS) feature selection competition. FSPP1-3 show good performance consistently with FSPP2 being the best overall by a slight margin. The performance of FSPP2 is competitive with some of the best performing feature-selection methods in the literature on the datasets that we have tested. Its associated computations are modest and hence it is suitable as a feature-selection method for SVM applications.

Keywords Feature selection · Feature ranking · Support vector machines · Sensitivity

1 Introduction

Feature selection is an important issue in machine-learning problems. When the underlying important features are known and irrelevant/redundant features are removed, learning problems can be greatly simplified resulting in improved generalization capabilities. Feature selection can also help reduce online computational costs, enhance system interpretability (Boser et al. 1992; Cortes and Vapnik 1995; Vapnik 1998) and improve performance of the learning problems (Saon and Padmanabhan 2001; Weston et al. 2001; Guyon et al. 2002; Günter and Bunke 2004 and others). Several feature-selection methods have been proposed in recent years and a good review of them can be found in the recent book by Guyon et al. (2006a). In general, feature-selection methods can be classified into three categories: filter-based, wrapper-based and embedded-based (Kohavi and John 1997; Guyon and Elisseeff 2003; Neumann et al. 2005). Filter-based methods are independent of the underlying learning algorithm while wrapper-based methods use the underlying learning algorithm to measure the quality of the features but without exploiting the structure of the learning algorithm. In contrast, embedded-based methods exploit the knowledge of the specific structure of the learning algorithm (Guyon and Elisseeff 2003; Lal et al. 2006) and cannot be separated from it. Generally, embedded-based methods are superior in performance relative to filter-based or wrapper-based methods but carry with them a heavier computational load (Guyon et al. 2006b).

This paper develops a new embedded-based feature-selection method specifically for Support Vector Machine (SVM) learning. The focus on SVM stems from the interests in it as a learning method following its encouraging results on a variety of applications (Boser et al. 1992; Cortes and Vapnik 1995; Vapnik 1995; Cristianini and Shawe-Taylor 2000). Unlike past feature-selection methods for SVM, this paper proposes the use of the probabilistic outputs of SVM as a more accurate measure of feature importance. For the prototypical two-class (c_1 and c_2) classification problem, probabilistic output of SVM for a given sample, $\mathbf{x}$, can be interpreted (Hastie and Tibshirani 1998; Platt 2000) as the posterior probability of $\mathbf{x}$ belonging to class c_1, $p(c_1|\mathbf{x})$. Such an interpretation under the Bayesian framework has also been established (Williams and Rasmussen 1996; Chu et al. 2003, 2004). This work proposes a criterion based on the sensitivity of probabilistic outputs of SVM to each feature as a measure of importance of that feature, and is termed Feature-based Sensitivity of Posterior Probabilities (FSPP). In loose terms, this criterion is the aggregate value, over the feature space, of the absolute difference of the probabilistic outputs of SVM with and without the feature.

The evaluation of this criterion is investigated using four approximations, termed FSPP1-FSPP4 respectively. These approximations are then combined with the recursive feature-elimination approach (Guyon et al. 2002) and other heuristic feature-ranking approaches to yield an overall feature-selection scheme. The first two approximations are motivated by the random forests feature-selection method (Breiman 1996, 2001) where the idea of Random Permutation (RP) of the values of a feature is used to eliminate the contribution of that feature. They differ from each other in that FSPP1 uses a simple threshold function to obtain the probabilistic output of SVM while FSPP2 uses a sigmoid function. The second two are direct approximations of the criterion. FSPP3 assumes mild dependence of the criterion with respect to the features while FSPP4 assumes that criterion is differentiable with respect to the features. The proposed methods are tested on several learning problems, including the MONK's problems, breast cancer and heart disease problems from the UCI Repository (Newman et al. 1998), the nonlinear synthetic problem of Weston et al. (2001) and another two challenging problems, ARCENE and MADELON, from the NIPS 2003 feature selection competition (Guyon et al. 2003). Numerical comparisons with two well-known existing SVM feature-selection methods (SVM-RFE by Guyon et al. 2002 and the margin method by Rakotomamonjy 2003) are also presented. The results show that FSPP2 performs consistently well on these datasets and compares favorably with the best methods available in the literature.

This paper is organized as follows. Past related results from the literature needed for the subsequent sections are collected in Sect. 2. Section 3 provides the basis of the proposed criterion and the descriptions of the four approximations of the criterion. Section 4 outlines the overall feature-selection schemes using the proposed criterion. Extensive experimental results are reported in Sect. 5, followed by discussion and future work in Sect. 6. The conclusions are drawn in Sect. 7.

2 Background

The section provides a review of standard SVM classifier, its probabilistic formulation and closely-related past work on SVM feature-selection methods. The intention is to set the notations for the remainder of this paper and to make the paper as self-contained as possible. We begin with the general notations used. This paper considers the typical two-class classification problem with dataset D in the form of $\{\mathbf{x}_j, y_j\}_{j=1}^N$ where $\mathbf{x}_j \in R^d$, is the jth sample and $y_j \in \{-1, 1\}$, the corresponding class label. Also, $\mathbf{x}^i$ denotes the ith feature of vector $\mathbf{x}$, hence, $\mathbf{x}_j^i$ is the ith feature of the jth sample and $\mathbf{x}_{-i} \in R^{d-1}$ is the vector obtained from $\mathbf{x}$ with the ith feature removed. Double subscripted variable $\mathbf{x}_{-i,j}$ is also used and it refers to the jth sample of variable $\mathbf{x}_{-i}$.

2.1 SVM classifier

Support Vector Machine is a well known learning method (Boser et al. 1992; Cortes and Vapnik 1995; Vapnik 1995; Cristianini and Shawe-Taylor 2000). Given a dataset D in the form of $\{\mathbf{x}_j, y_j\}_{j=1}^N$, standard SVM for the two-class classification problem maps the feature vector $\mathbf{x} \in R^d$ into a high (possibly infinite) dimensional Euclidean space, H, using a nonlinear mapping function $\Phi : R^d \to H$. The decision boundary of the two-class problems takes the form of an optimal separating hyperplane, $\mathbf{w} \cdot \Phi(\mathbf{x}) + b = 0$, in H, obtained by solving the convex optimization problem

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i, \\ &\text{s.t.} \quad y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) + \xi_i \geq 1 \quad \text{and} \quad \xi_i \geq 0, \quad \text{for } i = 1, \ldots, N, \end{aligned} \tag{1}$$

over $\mathbf{w} \in H, b \in R$ and the non-negative slack variable $\boldsymbol{\xi} \in R^N$. In the above, C is a parameter that balances the size of $\mathbf{w}$ and the sum of ξ_i. It is well known that the numerical computation of Problem (1) is achieved through its dual formulation. Suppose α_i be the Lagrange multiplier corresponding to the ith inequality, then the dual of (1) can be shown to be

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i, \\ &\text{s.t. } \sum_{i=1}^{N} y_i \alpha_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le C, \quad \text{for } i = 1, \dots, N, \end{aligned} \tag{2}$$

where the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ and

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{x}_i). \tag{3}$$

With (3), the expression of the hyperplane $\mathbf{w} \cdot \Phi(\mathbf{x}) + b = 0$ becomes

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{4}$$

and serves as the decision function for all unseen samples $\mathbf{x}$ in that the predicted class is $+1$ if $f(\mathbf{x}) > 0$ and -1 otherwise.

Many algorithms for the numerical solutions of (2) exist (Joachims 1999; Platt 1999; Chang and Lin 2001 and others) and several choices of the kernel function are available. For ease of presentation, the exposition hereafter uses, without loss of generality, the popular Gaussian kernel

$$K(\mathbf{x}_k, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_k - \mathbf{x}_j\|^2), \tag{5}$$

where γ is the kernel parameter. For accurate prediction of unseen samples, proper values of the parameters C and γ are used. Typically, these parameters are obtained using the cross-validation procedure although other methods have also been discussed (Lee and Lin 2000; Chapelle et al. 2002; Keerthi 2002).

2.2 Past work in SVM feature selection

Several feature-selection methods for SVM have been proposed in the literature (Bradley and Mangasarian 1998; Weston et al. 2001; Guyon et al. 2002; Rakotomamonjy 2003; Neumann et al. 2005). In most of these methods, the feature-ranking criterion relies on the sensitivity of some suitable index of performance, or its estimate, with respect to the feature. Features with low sensitivity are deemed less important while those with high sensitivity are more.

Index of performance is typically linked to generalization ability of SVM and several estimates of this ability have been used in the literature. Guyon et al. (2002) used the cost function of (1) and proposed a feature-ranking criterion based on the sensitivity of this cost function with respect to a feature. In loose terms, this criterion measures the importance of a feature by the difference in the sizes of the margin with and without the feature. For notational convenience, this criterion is denoted by $\Delta\|\mathbf{w}\|^2$ hereafter. Using this criterion as a basis, less important features are dropped successively, resulting in a feature-selection method known as SVM Recursive Feature-Elimination (SVM-RFE). Similarly, Weston et al. (2001) used, as the performance index, the SVM radius/margin bound (Vapnik 1998)

$$R^2\|\mathbf{w}\|^2, \tag{6}$$

where R is the radius of the smallest sphere, centered at the origin, that contains all $\Phi(\mathbf{x}_i)$, $i = 1, \dots, N$. The sensitivity of this index with respect to a feature was obtained through the use of a virtual scaling factor. As suggested by Weston et al. (2001), the idea could also be extended to the span estimate (Vapnik and Chapelle 2000) which is a tighter upper bound on the expected generalization error. Rakotomamonjy (2003) extended SVM-RFE algorithm using radius/margin bound and span estimate and proposed feature-selection methods based on their zero-order and first-order sensitivity with respect to the features. As reported (Rakotomamonjy 2003) to be the best among the considered methods, the first-order sensitivity, denoted by $\nabla\|\mathbf{w}\|^2$, is included in our numerical experiments for comparison.

2.3 Platt's probabilistic output

Standard SVM output classifies a sample $\mathbf{x}$ depending on the sign of $f(\mathbf{x})$, or the half space in H into which $\Phi(\mathbf{x})$ falls. Such an approach, however, ignores the relative confidence in the classification, or the distance $\Phi(\mathbf{x})$ is from the separating hyperplane. Platt (2000) addressed this shortcoming through the use of a sigmoid function and mapped $f(\mathbf{x})$ into $p(c|\mathbf{x})$, providing probabilistic information from standard SVM output. The benefit of $p(c|\mathbf{x})$ over $f(\mathbf{x})$ in improving classification accuracy has been demonstrated on several numerical experiments (Platt 2000; Duan and Keerthi 2005).

Suppose N_+ and N_- are the numbers of positive ($y = +1$) and negative ($y = -1$) samples respectively in dataset D. The Platt's probability output is

$$\hat{p}(c|\mathbf{x}) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)}, \tag{7}$$

where $f(\mathbf{x})$ is the SVM output given by (4) and the parameters A and B are obtained from minimizing the negative log likelihood (or the cross-entropy error function) of D in the form of

$$\min F(A, B) = \min\left\{-\sum_i [t_i \log \hat{p}(c|\mathbf{x}_i) + (1 - t_i)\log(1 - \hat{p}(c|\mathbf{x}_i))]\right\}, \tag{8}$$

with $t_i = (N_+ + 1)/(N_+ + 2)$ if $y_i = +1$ and $t_i = 1/(N_- + 2)$ if $y_i = -1$. Our implementation of the above includes the modifications suggested by Lin et al. (2003) for numerical stability. Hereafter, $\hat{p}(c|\mathbf{x})$ refers to the estimated posterior probability of belonging class $+1$ given $\mathbf{x}$ obtained from (7–8), while $p(c|\mathbf{x})$ refers to the true but typically unknown posterior probability of belonging to class c given $\mathbf{x}$. The quantity $\hat{p}(c|\mathbf{x})$ is used extensively in the approximations of the proposed ranking criterion.

3 The ranking criterion based on posterior probabilities

The proposed ranking criterion for the ith feature is

$$C_t(i) = \int |p(c|\mathbf{x}) - p(c|\mathbf{x}_{-i})|\,p(\mathbf{x})d\mathbf{x}, \tag{9}$$

where $\mathbf{x}_{-i} \in R^{d-1}$ is the vector derived from $\mathbf{x}$ with the ith feature removed. The motivation of the above criterion is clear: the greater the absolute difference between $p(c|\mathbf{x})$ and $p(c|\mathbf{x})_{-i})$ over the space of $\mathbf{x}$, the more important is the ith feature. As the true values of $p(c|\mathbf{x})$ and $p(c|\mathbf{x})_{-i})$ are usually unknown, they are approximated by $\hat{p}(c|\mathbf{x})$ and $\hat{p}(c|\mathbf{x}_{-i})$ respectively obtained via (7–8). The value of $\hat{p}(c|\mathbf{x}_{-i})$ corresponds to the probabilistic output of a SVM trained with data $\{\mathbf{x}_{-i,j}, y_j\}_{j=1}^{N}$ instead of $\{\mathbf{x}_j, y_j\}_{j=1}^{N}$. Since $\mathbf{x}$ has d features, this means that training of the SVM has to be done d times so that a ranked list of $\{C_t(i) : i = 1, \dots, d\}$ is obtained showing the relative importance of all features in D. This is a computationally expensive process since each SVM training is expensive, having a known complexity (Joachims 1999; Platt 1999) of at least $O(N^2)$ and that d can be large. The remainder of this section shows four approximations (FSPP1-FSPP4) of (9) that avoid the retraining process.

Motivated by the Random Forests (RF) method (Breiman 1996, 2001), the first two approximations involve a process of Random Permutation (RP) that randomly permutes the values of a feature. Specifically, the values of the ith feature of $\mathbf{x}$ are randomly permuted over the N examples. All other features of $\mathbf{x}$, except $\mathbf{x}^i$, remain unchanged. Suppose $\zeta_1, \dots, \zeta_{\mathrm{N}}$ is a set of uniformly distributed random numbers from (0,1) and $\lfloor \zeta \rfloor$ is the largest integer that is less than ζ. The random permutation process is executed as follows (Page 1967): For each k starting from 1 to $N-1$, compute $j = \lfloor N^* \zeta_k \rfloor + 1$ and swap the values of $\mathbf{x}_k^i$ and $\mathbf{x}_j^i$. At the end of this process, the values of $\mathbf{x}^i$ will have been randomly permuted.

We now state a general theorem relating the posterior probability and the RP process and it serves as the theoretical basis for FSPP1 and FSPP2. To state this theorem precisely, let $\mathbf{x}_{(i)} \in R^d$ be the vector derived from $\mathbf{x}$ with the ith feature randomly permuted.

Theorem 1

$$p(c|\mathbf{x}_{(i)}) = p(c|\mathbf{x}_{-i}). \tag{10}$$

Proof As the uniform distribution is used in the RP process, the distribution of $p(\mathbf{x}^i)$ is unchanged, or

$$p(\mathbf{x}_{(i)}^i) = p(\mathbf{x}^i). \tag{11}$$

Hence, we have

$$p(\mathbf{x}_{(i)}) = p(\mathbf{x}_{(i)}^i, \mathbf{x}_{-i}) = p(\mathbf{x}_{(i)}^i) p(\mathbf{x}_{-i}) = p(\mathbf{x}^i) p(\mathbf{x}_{-i}), \tag{12}$$

where the second equality follows from the fact that the distribution of the $p(\mathbf{x}_{(i)}^i)$ is independent from $p(\mathbf{x}_{-i})$ following the RP process. Using similar argument, we have

$$p(\mathbf{x}_{(i)}, c) = p(\mathbf{x}_{(i)}^i) p(\mathbf{x}_{-i}, c) = p(\mathbf{x}^i) p(\mathbf{x}_{-i}, c). \tag{13}$$

Hence,

$$p(c|\mathbf{x}_{(i)}) = \frac{p(c, \mathbf{x}_{(i)})}{p(\mathbf{x}_{(i)})} = \frac{p(\mathbf{x}^i) p(\mathbf{x}_{-i}, c)}{p(\mathbf{x}^i) p(\mathbf{x}_{-i})} = p(c|\mathbf{x}_{-i}). \quad \square \tag{14}$$

A corollary of Theorem 1 is the mutual information equality of $I(c, \mathbf{x}_{(i)}) = I(c, \mathbf{x}_{-i})$. This result follows from

$$\begin{aligned}
I(c,\mathbf{x}_{(i)}) &= \sum_c \int_{\mathbf{x}_{(i)}} p(c,\mathbf{x}_{(i)}) \log \frac{p(c,\mathbf{x}_{(i)})}{P(c)p(\mathbf{x}_{(i)})} d\mathbf{x}_{(i)} \\
&= \sum_c \int_{\mathbf{x}_{-i}} \int_{\mathbf{x}^i_{(i)}} p(\mathbf{x}^i_{(i)}) p(c,\mathbf{x}_{-i}) \log \frac{p(c,\mathbf{x}_{-i})}{P(c)p(\mathbf{x}_{-i})} d\mathbf{x}^i_{(i)} d\mathbf{x}_{-i} \\
&= \sum_c \int_{\mathbf{x}_{-i}} p(c,\mathbf{x}_{-i}) \log \frac{p(c,\mathbf{x}_{-i})}{P(c)p(\mathbf{x}_{-i})} d\mathbf{x}_{-i} \int_{\mathbf{x}^i_{(i)}} p(\mathbf{x}^i_{(i)}) d\mathbf{x}^i_{(i)} \\
&= I(c,\mathbf{x}_{-i}),
\end{aligned} \tag{15}$$

where (12) and (13) are invoked.

Theorem 1 and its corollary show that the RP process has the same effect as removing the contribution of that feature for classification. Using this fact, criterion (9) can be equivalently stated as

$$C_t(i) = \int |p(c|\mathbf{x}) - p(c|\mathbf{x}_{(i)})| p(\mathbf{x}) d\mathbf{x}. \tag{16}$$

With (16), we are now in a position to state the first two approximations of the proposed ranking criterion.

Method 1 (FSPP1): *Approximation using threshold function*
The first method uses a threshold function for the approximation of (16) in the form of

$$p(c|\mathbf{x}) \approx \varphi(f(\mathbf{x})) \tag{17}$$

and

$$p(c|\mathbf{x}_{(i)}) \approx \varphi(f(\mathbf{x}_{(i)})), \tag{18}$$

where $\varphi(\cdot)$ is the threshold function given by

$$\varphi(f) = \begin{cases} 1 & \text{if } f \geq 0, \\ 0 & \text{if } f < 0. \end{cases} \tag{19}$$

It is worthy to note that $p(c|\mathbf{x}_{(i)})$ uses the same f function as in (17) and does not involve the retraining of the SVM. Further approximation of the integration over $\mathbf{x}$ in (16) yields

$$\text{FSPP1}(i) = \frac{1}{N} \sum_{j=1}^{N} |\varphi(f(\mathbf{x}_j) - \varphi(f(\mathbf{x}_{(i),j}))|, \tag{20}$$

where $\mathbf{x}_{(i),j}$ refers to the jth example of the input data where the ith feature has been randomly permuted.

Method 2 (FSPP2): *Approximation using SVM probabilistic outputs*
Motivated by the good results reported by Platt (2000), Duan and Keerthi (2005), FSPP2 approximates $p(c|\mathbf{x})$ by the Platt's probabilistic output, $\hat{p}(c|\mathbf{x})$, in (16). Obviously, other methods that obtain probabilistic outputs from SVM can also be used (Vapnik 1998; Hastie and Tibshirani 1998). Similarly, $p(c|\mathbf{x}_{(i)})$ in (16) is approximated by $\hat{p}(c|\mathbf{x}_{(i)})$ using the same trained SVM and the same trained sigmoid for $\hat{p}(c|\mathbf{x})$. Hence,

$$\text{FSPP2}(i) = \frac{1}{N} \sum_{j=1}^{N} |\hat{p}(c|\mathbf{x}_j) - \hat{p}(c|\mathbf{x}_{(i),j})|. \tag{21}$$

Method 3 (FSPP3): *Approximation via virtual vector* $\mathbf{v}$

Unlike the previous, the next two methods (FSPP3 and FSPP4) approximate (9) via an additional virtual scaling factor. The use of an additional virtual vector $\mathbf{v} \in R^d$ for the purpose of feature selection has been attempted in the literature (Weston et al. 2001; Rakotomamonjy 2003) and it simplifies the computation of (9). Specifically, this approach uses one $\mathbf{v}^i$, having a nominal value of 1, for each feature and replaces every $\mathbf{x}^i$ by $\mathbf{v}^i\mathbf{x}^i$. Let $\mathbf{vx} = [\mathbf{v}^1\mathbf{x}^1\ \mathbf{v}^2\mathbf{x}^2\ \dots\ \mathbf{v}^d\mathbf{x}^d]^T$ and $\mathbf{v}_{-i}\mathbf{x}$ refers to $\mathbf{vx}$ with $\mathbf{v}^i = 0$. In this setting, the criterion (9) can be approximated by

$$C_t(i) = \int |p(c|\mathbf{vx}) - p(c|\mathbf{v}_{-i}\mathbf{x})| p(\mathbf{x}) d\mathbf{x}. \tag{22}$$

Using standard approximation, the above becomes

$$\text{FSPP3}(i) = \frac{1}{N}\sum_{j=1}^{N} |\hat{p}(c|\mathbf{vx}_j) - \hat{p}(c|\mathbf{v}_{-i}\mathbf{x}_j)|, \tag{23}$$

where $\hat{p}(c|\mathbf{vx}_j)$ refers to the Platt's posterior probability of the jth example and $\hat{p}(c|\mathbf{v}_{-i}\mathbf{x}) = (1 + \exp(Af(\mathbf{v}_{-i}\mathbf{x}) + B))^{-1}$ as given by (7) and $f(\cdot)$ is the SVM output expression (4) obtained from the training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$.

Method 4 (FSPP4): *Approximation via derivative of* $p(c|\mathbf{vx})$ *with respect to* $\mathbf{v}$

The criterion of (22) can also be represented, under the assumption that $p(c|\mathbf{vx})$ is differentiable with respect to $\mathbf{v}$, by

$$C_t(i) = \int \left| \int_{\mathbf{v}^i=1}^{\mathbf{v}^i=0} \frac{\partial p(c|\mathbf{vx})}{\partial \mathbf{v}^i} d\mathbf{v}^i \right| p(\mathbf{x}) d\mathbf{x}. \tag{24}$$

Instead of the integral over $\mathbf{v}^i$ from 1 to 0, FSPP4 uses the sensitivity with respect to $\mathbf{v}^i$ evaluated at $\mathbf{v}^i = 1$ and (24) is approximated by

$$C_t(i) = \int \left| \frac{\partial p(c|\mathbf{vx})}{\partial \mathbf{v}^i} \Delta\mathbf{v}^i |_{\mathbf{v}^i=1} \right| p(\mathbf{x}) d\mathbf{x} = \int \left| \frac{\partial p(c|\mathbf{vx})}{\partial \mathbf{v}^i} |_{\mathbf{v}^i=1} \right| p(\mathbf{x}) d\mathbf{x}, \tag{25}$$

where $\Delta\mathbf{v}^i = -1$. It is important to note that, when $p(c|\mathbf{x})$ is approximated by $\hat{p}(c|\mathbf{x})$ of (7), $\partial\hat{p}(c|\mathrm{v}x)/\partial v^i$ admits a closed-from expression using the results of (4) and (7). This expression and its derivation are given in appendix. Hence, the fourth method is

$$\text{FSPP4}(i) = \frac{1}{N}\sum_{j=1}^{N} \left| \frac{\partial \hat{p}(c|\mathbf{vx}_j)}{\partial \mathbf{v}^i} |_{\mathbf{v}^i=1} \right|. \tag{26}$$

The above shows four possible approximations to (9). The use of these four methods, in an overall scheme for the purpose of feature selection, is shown next.

4 Feature-selection methods

This section presents two overall feature-selection schemes by combining FSPP1-FSPP4 with either the initial feature-ranking (INIT) approach (FSPP-INIT) or the recursive feature-elimination (RFE) approach (FSPP-RFE). Both INIT and RFE approaches are commonly used for feature selection, with INIT being closer to the filter-based method and the RFE being closer to the embedded method (Guyon and Elisseeff 2003; Guyon et al. 2006a).

For both of the proposed feature-selection schemes (FSPP-INIT and FSPP-RFE), it is assumed that an SVM output function $f(\mathbf{x})$ is available and that all hyper parameters, C, γ or others, have been determined through a proper model selection process. For the cases where FSPP2-FSPP4 are involved, it is also assumed that the posterior probabilities are available according to (7) and (8).

The FSPP-INIT scheme has as its inputs dataset D, the index set $I = \{1, 2, \dots, d\}$ containing indices of features to be considered and the choice of the approximation method $m \in \{1, \dots, 4\}$. The output of FSPP-INIT is a ranked list of the features in the form of an index set $J_m = \{j_1, j_2, \dots, j_d\}$ with $j_k \in I$ and FSPP$m(j_k) \geq$ FSPP$m(j_{k+1})$ for $k = 1, \dots, d-1$.

```
FSPP-INIT($D, I, m$):
1. For each $i \in I$, compute FSPP$m(i)$ using the data set $D$.
2. Output ranked list $J_m$.
```

The FSPP-RFE scheme is similar to the one given by Guyon et al. (2002) but with the FSPPm used as the ranking criterion. The steps involved in this approach are summarized as follows. The inputs are the dataset D and m, with the output being the ranked list of features J_R.

```
FSPP-RFE($D$, $m$):
1. Let $I = \{1, 2, \dots, d\}$ and $\ell = d$.
2. If $I = \emptyset$, stop. Else, invoke FSPP-INIT($D, I, m$) and obtain
   the output $J_m$.
3. Let the last element of $J_m$ be $\hat{k}$. Assign $\hat{k}$ to the $\ell$th
   element of $J_R$.
4. Let $I = I \backslash \hat{k}, \ell = \ell - 1$ and remove feature $\hat{k}$ from every sample
   in $D$.
5. Retrain SVM with $D$ and obtain the posterior probabilities
   using (7) and (8). Goto 2.
```

As the FSPP-INIT scheme computes the ranked list only once, it is closer in spirit to a filter-based feature-selection scheme although the SVM algorithm is used. On the other hand, the FSPP-RFE scheme uses FSPP-INIT as an inner-loop and invokes it $d - 1$ times, each time with a smaller index set I. Steps 3 and 4 of FSPP-RFE(D, m) above remove one feature (the one with the lowest FSPPm score) from the dataset at a time. Obviously, more than one feature can be removed at one time with slight modifications to Steps 3 and 4. The current description of FSPP-RFE does not involve the determination of parameters C and γ for each of the inner loop. Such a process is possible albeit with even higher costs. For notational convenience, FSPPm-INIT and FSPPm-RFE are used to specify the feature selection scheme using FSPPm as the choice of the approximation method.

5 Experiments

Extensive experiments on both artificial and real-world benchmark problems were carried out using the proposed methods. Like others, the artificial problems, i.e. MONK's problems from UCI Repository (Newman et al. 1998) and Weston's nonlinear synthetic problem (Weston et al. 2001), were used because the key features are known and are suitable for comparative study of the four FSPPs. Two real-world problems, i.e. breast cancer and heart disease problems from UCI Repository (Newman et al. 1998; Rätsch 2005), were chosen as they have been used by other feature-selection methods (Guyon et al. 2002; Rakotomamonjy 2003) and serve as a common reference for comparison. Finally, the proposed methods were tested on ARCENE and MADELON problems used in the NIPS 2003 feature selection competition (Guyon et al. 2003), a well-known set of challenging feature-selection problems.

In general, our method requires, for each problem, three subsets of data in the form of D_{tra}, D_{val} and D_{tes} for training, validation and testing purposes. In cases where only D_{tra} and D_{tes} were available, D_{tra} was further split randomly into a new D_{tra} and D_{val} in the ratio of 70% to 30%. The subset D_{tra} was normalized to zero mean and unit standard deviation. Its normalizing parameters were also used to normalize D_{val} and D_{tes}. The subset D_{tra} was meant for the training of the SVM including the determination of the optimal C and γ using 5-fold cross-validation procedure. The subset D_{val} was needed for the determination of parameters A and B in (7). The D_{tes} subset was used for obtaining an unbiased testing accuracy of the underlying method. In cases where there were 100 realizations of a given dataset, the procedure by Rätsch et al. (2001) was followed: parameters C and γ were chosen as the median of the five sets of (C,γ) of the first five realizations. Here each set of (C,γ) was obtained by standard 5-fold cross-validations for one realization.

5.1 Artificial problems

MONK's problems These problems (MONK-1 to 3) are available in UCI Repository of machine learning databases (Newman et al. 1998). As the provided data do not have D_{val} and the size of D_{tra} is relatively small, our experiments used part of the test set to form D_{val} and D_{tra}. The exact data split and the descriptions of the dataset are given in Table 1.

The results for MONK-1 experiment using the optimal parameters ($C = 32$ and $\gamma = 0.125$) are shown in Fig. 1. Figure 1(a) shows the FSPP*m* scores for the four methods using the INIT approach. It is easy to see that all four methods were effective in determining the key features. Figure 1(b) shows the test error rates of SVM using only the top-ranked features obtained via the RFE approach. The monotonic decrease in the testing error rates with increasing top-ranked features is a clear indication of the effectiveness of the feature-selection procedure. The results for MONK-2 and MONK-3 show similar trends to Fig. 1 and are hence not shown.

The test error rates for FSPP4-RFE are not shown in Fig. 1(b) as the computation of (31) failed. This problem arose due to the existence of multiple identical examples in the training data, resulting in the matrix in (31) being singular. While less likely to occur in real-life datasets, such situations can be handled using pseudo inverses and/or Singular Value Decomposition (SVD) of the matrix in (31). However, they were not pursued because the performance of FSPP4 for other examples is not promising, as shown in the next few examples.

Weston's nonlinear synthetic problem We followed the procedure given in (Weston et al. 2001) and generated 10,000 samples of 10 features each. Only the first two features ($\mathbf{x}^1$, $\mathbf{x}^2$) are relevant while the remaining features are random noise, each taken from a normal distribution $N(0, 20)$. The output $y \in \{-1, +1\}$ and the number of samples with $y = +1$ is equal to that with $y = -1$. If $y = -1$, ($\mathbf{x}^1$, $\mathbf{x}^2$) were drawn from $N(\boldsymbol{\mu}_1, \sum)$ or $N(\boldsymbol{\mu}_2, \sum)$ with equal probability, with $\boldsymbol{\mu}_1 = (-3/4, -3)$, $\boldsymbol{\mu}_2 = (3/4, 3)$ and $\sum = \boldsymbol{I}$. If $y = +1$, ($\mathbf{x}^1$, $\mathbf{x}^2$) were drawn again from two normal distributions with equal probability, with $\boldsymbol{\mu}_1 = (3, -3)$, $\boldsymbol{\mu}_2 = (-3, 3)$ and the same $\sum$. D_{tra} and D_{val} contained 100 random samples each and the rest were included in D_{tes} for one realization of the dataset.

Table 1 Description of MONK's datasets (Five discrete features: $x_1, x_2, x_4 \in \{1,2,3\}$; $x_3, x_6 \in \{1,2\}$; $x_5 \in \{1,2,3,4\}$)

	D_{tra}	D_{val}	D_{tes}	Target concept
MONK-1	216	216	124	$(x_1 = x_2)$ or $(x_5 = 1)$ for Class 1, otherwise Class -1
MONK-2	216	216	169	Exactly two of $\{x_1 = 1, x_2 = 1, x_3 = 1,$ $x_4 = 1, x_5 = 1, x_6 = 1\}$ for Class 1, otherwise Class -1
MONK-3	216	216	122	$(x_5 = 3$ and $x_4 = 1)$ or $(x_5 \neq 4$ and $x_2 \neq 3)$ for Class 1, otherwise Class -1

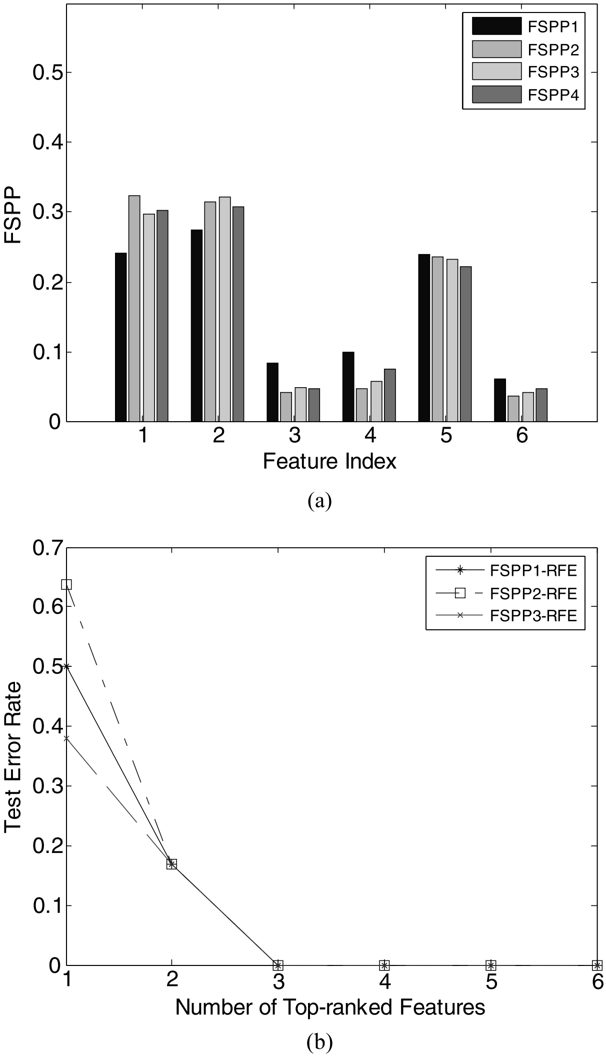
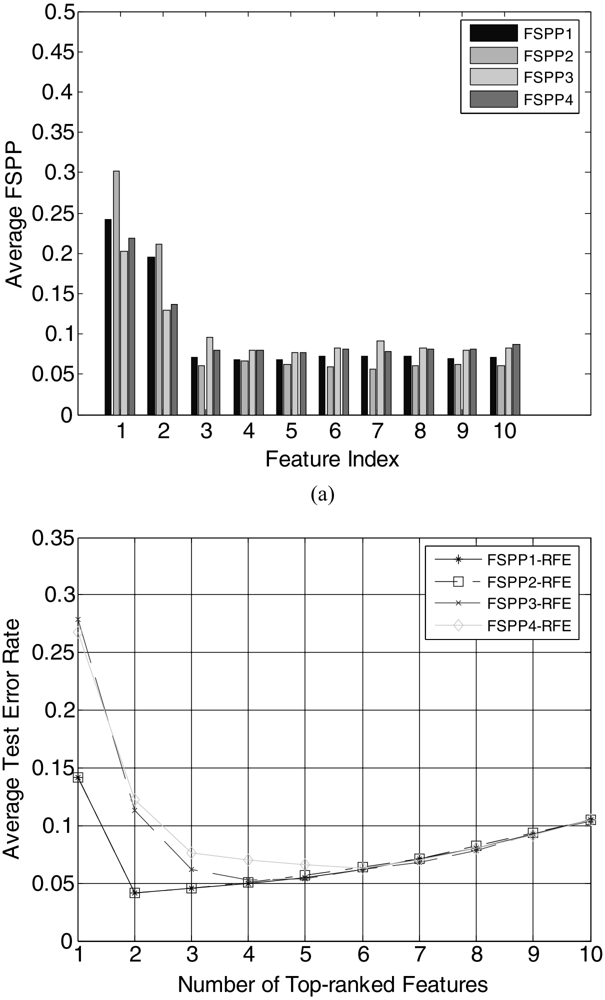


Fig. 1 Performance of proposed methods on MONK-1 problem: **a** values of FSPPm, $m = 1, 2, 3, 4$ using FSPPm-INIT; **b** test error rates against top-ranked features identified by FSPPm-RFE. **a** shows the effectiveness of the FSPP1-4 in identifying the key features (features 1, 2 and 5) under the INIT approach as key features were assigned larger FSPPm scores. **b** shows the monotonic decrease in the test error rates with increasing top-ranked features indicating the effectiveness of the proposed methods in identifying the more important features using the RFE approach. The test error rates for FSPP4-RFE are not shown in **b** as the computation of (31) failed during RFE process. See description in Sect. 5.1

Average feature-selection performance over 100 realizations is shown in Fig. 2 with the parameters set at $C = 32.0, \gamma = 0.03125$. Similar to the MONK's problems, Fig. 2(a) and (b) were obtained from the use of FSPPm-INIT and FSPPm-RFE respectively. Fig-



(a)

(b)

Fig. 2 Performance of the proposed methods on Weston's nonlinear dataset: **a** values of FSPPm, $m = 1, 2, 3, 4$ using FSPPm-INIT; **b** test error rates against top-ranked features identified by FSPPm-RFE. Note that the stated FSPPm values and test error rates are the averages over 100 realizations. **a** shows that the key features (feature 1, 2) have larger FSPPm scores compared to those of the redundant features (P-values < 0.01 based on paired t-tests over the 100 realizations) using the INIT approach. **b** shows that FSPP1 and FSPP2 yielded significantly lower average test error rates than FSPP3 and FSPP4 using the RFE approach with P-values < 0.03 for paired t-tests. See Sect. 5.1 for details

ure 2(a) shows the correct identification of the first two features having FSPPm scores that are significantly larger (P-value < 0.01 based on paired t-test over the 100 realizations) than the FSPPm scores of a redundant feature. Figure 2(b) shows that FSPP1-RFE and FSPP2-RFE correctly identified the two key features as the test error rates were the lowest with only two surviving features. However, FSPP3-RFE and FSPP4-RFE produced less appealing results. Additional experiments were conducted to verify the statistical significance of the advantage of FSPP1 and FSPP2 over FSPP3 and FSPP4 under the RFE approach. Four paired t-tests on the test error rates were conducted: FSPP1 vs FSPP3, FSPP1 vs FSPP4, FSPP2 vs FSPP3 and FSPP2 vs FSPP4. Each of these t-tests was further repeated with only 1, 2, 3 or 4 surviving features. For all of these paired t-tests, the P-values obtained were less than 0.03.

The difference between the performance of FSPP2 and FSPP3 is interesting and deserves attention. Both criteria use the same $\hat{p}$ expression obtained from (7) and (8) but differ in that $\hat{p}(c|\mathbf{x}_{(i),j})$ is used in FSPP2 and $\hat{p}(c|\mathbf{v}_{-i}\mathbf{x}_j)$ in FSPP3. The sample $\mathbf{x}_{(i),j}$ has the ith feature taking value that is randomly permuted while $\mathbf{v}_{-i}\mathbf{x}_j$ has the ith feature set to 0. The better performance of FSPP2 over FSPP3 appears to suggest that the distribution $\hat{p}(c|\mathbf{v}_{-i}\mathbf{x})$ differs more from $p(c|\mathbf{x}_{-i})$ than $\hat{p}(c|\mathbf{x}_{(i)})$.

5.2 Real-world benchmark problems

The real-world benchmark problems are the breast cancer and heart disease datasets obtained from Rätsch (2005), used also by Mika et al. (1999), Rätsch et al. (2001) and Rakotomamonjy (2003) in their experiments. Sizes of feature/$D_{\text{tra}}/D_{\text{val}}/D_{\text{tes}}$ are 9/140/60/77 and 13/119/51/100 respectively and each problem has 100 realizations. For comparison purposes, the format of presentation of results by Rakotomamonjy (2003) was adopted. Plots of the mean test error rates of SVM are provided with decreasing number of top-ranked features. Each plot is the mean over 100 realizations using either FSPP-RFE or FSPP-INIT feature-selection scheme.

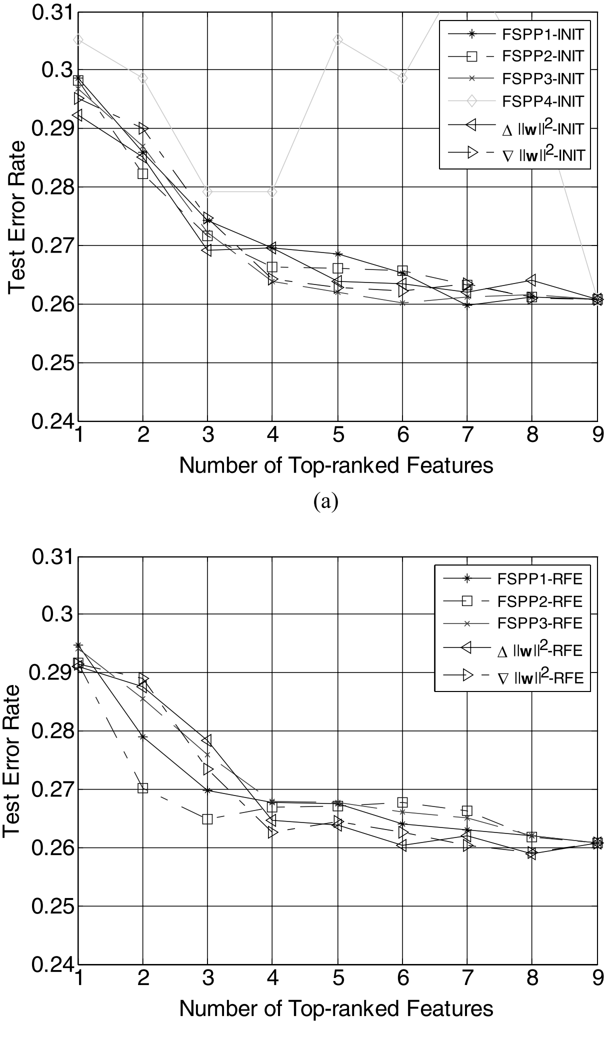
For comparison purposes, performance of two feature-ranking criteria, the $\Delta\|\mathbf{w}\|^2$ method by Guyon et al. (2002) and the $\nabla\|\mathbf{w}\|^2$ method by Rakotomamonjy (2003), is also included. They were chosen because they appear to have performed well (Rakotomamonjy 2003; Weston et al. 2001). Their performance was reproduced together with those using FSPP1-4 in Figs. 3 and 4 for the two problems. While Fig. 3 is for breast cancer dataset and Fig. 4 is for the heart disease dataset, Figs. 3(a) and 4(a) report on the results based on the INIT approach while Figs. 3(b) and 4(b) are results of the RFE approach. These results were obtained for the optimal parameters: ($C = 2.83$, $\gamma = 0.05632$) for the breast cancer dataset and ($C = 2.38$, $\gamma = 0.00657$) for the heart disease dataset.

Under the INIT approach, Fig. 3(a) shows that all the methods considered (except FSPP4) produced similar test error rates for the breast cancer dataset. This is confirmed by the P-values (>0.05) obtained from paired t-tests for the 100 realizations, except for FSPP4 which gave P-values of less than 0.01 when compared to other methods. This was, however, not observed for the heart disease dataset. Figure 4(a) shows that the FSPP1-4 are significantly better than the $\Delta\|\mathbf{w}\|^2$ and the $\nabla\|\mathbf{w}\|^2$ methods with P-values being less than 0.01 in the paired t-tests for FSPPm vs $\Delta\|\mathbf{w}\|^2$ and FSPPm vs $\nabla\|\mathbf{w}\|^2$. The performance of FSPP4 is not appealing for the breast cancer data. One possible reason is that the function $\hat{p}(c|\mathbf{v}\mathbf{x})$ as a function of $\mathbf{v}^i$ is highly nonlinear and not well approximated by $\partial\hat{p}(c|\mathbf{v}\mathbf{x})/\partial\mathbf{v}^i$ evaluated at $\mathbf{v}^i = 1$ as in (26).

For the RFE approach, Fig. 3(b) shows that FSPP1 and FSPP2 again yielded significantly lower average test error rates than FSPP3, $\Delta\|\mathbf{w}\|^2$ and $\nabla\|\mathbf{w}\|^2$. This is confirmed by the paired t-tests with P-values < 0.05 when only the top 2 or 3 features were used. Figure 3(b) further shows that FSPP2 had a slight edge over FSPP1 and produced lower average test error rates when only the top 2 or 3 features were used (P-values < 0.05), suggesting that FSPP2 could be the best performing method. In Fig. 4(b), the advantage of the FSPPm over the other two methods is obvious. The paired t-tests between FSPPm versus either of the two methods yielded P-values of less than 0.03. The variation in performance among FSPP1-3 are, however, not significant as the P-values were greater than 0.05. Also, FSPP4-RFE is not shown in Fig. 3(b) or Fig. 4(b) as the computation of (31) failed during the recursive feature elimination process.

5.3 NIPS challenge problems

A well-known set of challenging feature-selection problems is that given in the NIPS challenge problems (Guyon et al. 2003). These problems are known to be difficult and are designed to test various feature-selection methods using an unbiased testing procedure without revealing the labels of the test set. The problem sets ARCENE and MADELON were



(a)

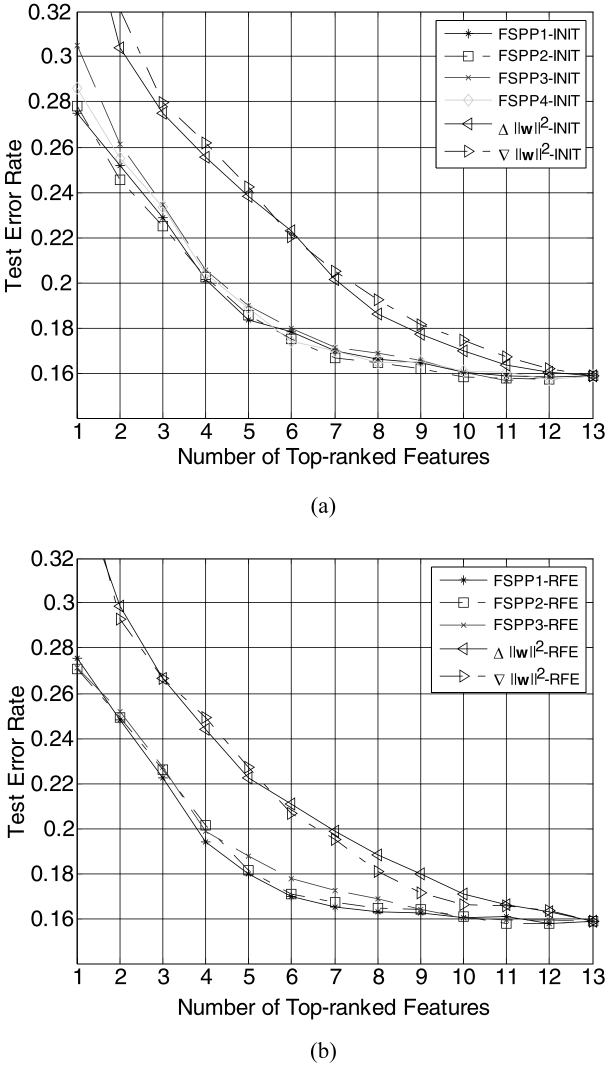
(b)

Fig. 3 Test error rates against top-ranked features on breast cancer dataset where the top-ranked features were chosen based on **a** FSPPm-INIT **b** FSPPm-RFE, $m = 1, 2, 3, 4$. Results of two other methods, $\Delta\|\mathbf{w}\|^2$ and $\nabla\|\mathbf{w}\|^2$, were also included. The test error rates shown are the averages over 100 realizations. **a** shows that FSPP1-3, $\Delta\|\mathbf{w}\|^2$ and $\nabla\|\mathbf{w}\|^2$ produced similar test error rates (P-values > 0.05). The performance of FSPP4 is not as appealing as the other methods (P-values < 0.01). **b** shows that FSPP1 and FSPP2 again yielded significantly lower average test error rates than FSPP3, $\Delta\|\mathbf{w}\|^2$ and $\nabla\|\mathbf{w}\|^2$ in the recursive feature elimination process, especially when fewer top-ranked features were used (P-values < 0.05 when only 2 or 3 top-ranked features were used). **b** further shows that FSPP2 had a slight edge over FSPP1 when only the top 2 or 3 features were used (P-values < 0.05), suggesting that FSPP2 could be the best performing method. See discussion in Sect. 5.2 for details

chosen to evaluate our proposed method. In view of time and space constraints, only the results of FSPP2-RFE are reported. The details of the ARCENE and MADELON datasets are given in Table 2. ARCENE is probably the most challenging among all the datasets from the NIPS competition as it is a sparse problem with the smallest examples-to-features ratio (num-of-training-examples/num-of-features = 100/10000), while MADELON is a relatively easier problem with a bigger examples-to-features ratio (2000/500). They were chosen to show effectiveness of the proposed methods for both sparse and non-sparse problems.

Based on the results of the earlier experiments, FSPP2-RFE was chosen for these two datasets. Our version of FSPP2-RFE used a three-tier removal of features for MADELON: 100 features at each recursion until 100 features were left followed by 20 features at each recursion until 20 features were left and finally one feature at each recursion. A more aggressive removal scheme was used for ARCENE: 1000 features were deleted at each recursion.

Fig. 4 Test error rates against top-ranked features on heart disease dataset where the top-ranked features were chosen based on **a** FSPPm-INIT **b** FSPPm-RFE, $m = 1, 2, 3, 4$. Results of two other methods, $\Delta\|\mathbf{w}\|^2$ and $\nabla\|\mathbf{w}\|^2$, were also included. The test error rates shown are the average over 100 realizations. The figures show that FSPP1-4 outperformed $\Delta\|\mathbf{w}\|^2$ and $\nabla\|\mathbf{w}\|^2$ (P-values < 0.03). The test error rates for FSPP4-RFE are not shown in **b** as the computation of (31) failed during the recursive feature elimination process. See discussion in Sect. 5.2 for details



(a)

(b)

Table 2 Description of ARCENE and MADELON datasets

Dataset	Features	D_{tra}	D_{val}	D_{tes}
MADELON	500	2000	600	1800
ARCENE	10000	100	100	700

For each dataset, our result of FSPP2-RFE having the best validation accuracy was chosen. Our entries were respectively ranked 1st and 2nd (as of February 01, 2006) in the MADELON and ARCENE group of entries. A comparison between our results and the best entries by other participants of the challenge (see Guyon et al. 2003) is given in Table 3 (as of February 01, 2006).

Table 3 Results on NIPS 2003 challenge datasets as of February 01, 2006

Dataset	Our best entry by FSPP2-RFE					Top entry by other researchers				
	Rank	BER	AUC	Feat. No.	Probe (%)	Rank	BER	AUC	Feat. No.	Probe (%)
MADELON	1	0.0622	0.9378	12	0.00	2	0.0622	0.9807	500	96
ARCENE	2	0.1060	0.8940	5000	27.82	1	0.0720	0.9811	100	0.00

BER is the balanced error rate on D_{tes}, while AUC refers to area under the ROC curve

6 Discussion and future work

In summary, FSPP1-3 performed well for all the artificial datasets. This is to be expected of any good feature-selection method. For the real-world datasets, FSPP1-2 had better performance than FSPP3 with the edge going to FSPP2, especially when small numbers of top-ranked features were used. The excellent performance of FSPP2 in the two NIPS challenge problems reaffirmed its suitability for real-world datasets.

FSPP2-RFE appears to do well on sparse datasets (datasets with large number of features but small training samples), as seen in the experiment associated with the ARCENE problem. The reason for its good performance is not exactly clear, but one possible reason is that the FSPP2 is based on the ensemble of all training examples of $|p(c|\mathbf{x}) - p(c|\mathbf{x}_{-i})|$ over the feature space, as seen in (9). This ensemble over all $\mathbf{x}_i$ is likely to be more accurate in measuring the contribution of a feature and is more robust against decreasing training examples. This is different from other methods that rely on bounds of index of performance where many of these bounds are known to be loose (Vapnik 1998; Rakotomamonjy 2003) and its effect could be more severe when the ratio of samples-to-features is low.

One significant advantage of FSPP2 is the modest computations needed for its evaluation. Suppose the SVM output $f(\mathbf{x}_i)$ is available for all $\mathbf{x}_i$ in the training data. The evaluation of $\hat{p}(c|\mathbf{x})$ requires a one-time determination of variables A and B from the optimization problem (8). Since (8) is an unconstrained convex optimization problem in two variables, its numerical determination is straight forward (Lin et al. 2003). The random permutation of every feature over the training data is required and it is a simple $O(dN)$ operation which can be done efficiently. Hence, the FSPP2 scales linearly with respect to the number of features or training samples and is suitable for large problems in high dimensions.

The proposed idea of using sensitivity of posterior probabilities for feature selection appears general and should be extendable to other machine learning algorithms where probabilistic outputs are also available.

The idea of using sensitivity of posterior probabilities for feature selection has been demonstrated in the context of two-class classification problem. Possible extensions of the current work could include the adaptation of the criterion to regression problems and multiclass classification problems where feature selection methods remain rare in the literature.

7 Conclusions

This paper introduces a new feature-ranking criterion based on the posterior probability of the SVM output. It is motivated from the advantage gained in using posterior probability as a decision function for classification instead of the direct SVM output function. Four approximations, with some motivated by the random-forests feature-selection method, are proposed for the evaluations of the criterion. These approximations are used in two overall feature-selection approaches, recursive feature-elimination approach and initial feature-ranking approach.

The experimental results on various datasets show that three of the four approximations (FSPP1, 2 and 3) yield good overall performance under the recursive feature-elimination approach. Among them, FSPP2 has the overall edge in terms of accuracy and shows performance that is comparable with some of the best methods in the literature. In addition, FSPP2 has modest computation and hence, is suitable for large problems in high dimensional feature space. In addition, it appears to perform well for datasets with low samples-to-features ratios. Consequently, this method is a good candidate for feature selection for SVM applications.

Appendix

This appendix shows the derivation of $\partial \hat{p}(c|\mathbf{vx}_j)/\partial \mathbf{v}^i$ used in (26) of FSPP4. Let $\hat{p}_j$, f_j denote $\hat{p}(c|\mathbf{vx}_j)$ and $f(\mathbf{vx}_j)$ respectively. Suppose there are m support vectors after the training/tuning of SVM. Let $I_1 = \{k|0 < \alpha_k < C\}$ and $I_2 = \{k|\alpha_k = C\}$ with cardinalities m_1 and m_2 respectively with $m_1 + m_2 = m$. From (4), (5) and (7), it is easy to see that

$$\left.\frac{\partial \hat{p}_j}{\partial \mathbf{v}^i}\right|_{\mathbf{v}^i=1} = -\frac{\exp(Af_j+B)}{[1+\exp(Af_j+B)]^2}\left.\left[A\frac{\partial f_j}{\partial \mathbf{v}^i} + f_j\frac{\partial A}{\partial \mathbf{v}^i} + \frac{\partial B}{\partial \mathbf{v}^i}\right]\right|_{\mathbf{v}^i=1}, \tag{27}$$

with

$$\frac{\partial f_j}{\partial \mathbf{v}^i} = \sum_{k=1}^{m}[(-2\gamma)\alpha_k y_k(\mathbf{x}_{k,i} - \mathbf{x}_{j,i})^2 K(\mathbf{vx}_k, \mathbf{vx}_j) + y_k K(\mathbf{vx}_k, \mathbf{vx}_j)\partial\alpha_k/\partial\mathbf{v}^i] + \partial b/\mathbf{v}^i. \tag{28}$$

Expression of the 1st term in the RHS of (27) involves the evaluations of $\partial\alpha_k/\partial\mathbf{v}^i$ for $k \in I_1$ and $\partial b/\partial\mathbf{v}^i$ as shown in (28), where the mild assumption of $\partial\alpha_k/\partial\mathbf{v}^i = 0$ for $k \in I_2$ is used. Using the Karush–Kuhn–Tucker (KKT) conditions (Cristianini and Shawe-Taylor 2000) of the SVM solutions, it is not difficult to show that

$$\begin{cases} \sum\limits_{k\in I_1}\alpha_k y_k K(\mathbf{vx}_k, \mathbf{vx}_p) + \sum\limits_{k\in I_2}\alpha_k y_k K(\mathbf{vx}_k, \mathbf{vx}_p) + b = y_p, \quad \forall p \in I_1, \\ \sum\limits_{k\in I_1}\alpha_k y_k + \sum\limits_{k\in I_2}\alpha_k y_k = 0, \end{cases} \tag{29}$$

or

$$\begin{bmatrix}\mathbf{A} & \mathbf{e}\\ \tilde{\mathbf{y}}^T & 0\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\alpha}}\\ b\end{bmatrix} + \begin{bmatrix}\boldsymbol{\beta}\\ \beta_0\end{bmatrix} = \begin{bmatrix}\tilde{\mathbf{y}}\\ 0\end{bmatrix}, \tag{30}$$

where $\mathbf{A}_{pk} = y_k K(\mathbf{vx}_k, \mathbf{vx}_p)$, $\tilde{\mathbf{y}}$ is the vector of $\mathrm{y}_i (i \in I_1)$, $\mathbf{e}$ is $m_1 \times 1$ vector of all 1, $\tilde{\boldsymbol{\alpha}}$ is the vector of $\alpha_i (i \in I_1)$, $\beta_0 = \sum_{k\in I_2}\alpha_k y_k$ and $\boldsymbol{\beta}_p = \sum_{k\in I_2}\alpha_k y_k K(\mathbf{vx}_k, \mathbf{vx}_p)$. Differentiate (30) with respect to $\mathbf{v}^i$ yields

$$\begin{bmatrix}\frac{\partial\tilde{\boldsymbol{\alpha}}}{\partial\mathbf{v}^i}\\ \frac{\partial b}{\partial\mathbf{v}^i}\end{bmatrix} = -\begin{bmatrix}\mathbf{A} & \mathbf{e}\\ \tilde{\mathbf{y}}^T & 0\end{bmatrix}^{-1}\left\{\begin{bmatrix}\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{v}^i}\\ 0\end{bmatrix} + \begin{bmatrix}\frac{\partial\mathbf{A}}{\partial\mathbf{v}^i} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\alpha}}\\ b\end{bmatrix}\right\}. \tag{31}$$

The 2nd and 3rd terms in the RHS of (27) involve differentiations of A and B. From (8), the solutions for A and B have to satisfy

$$\frac{\partial F(A,B)}{\partial A} = -\sum_j \left(\frac{t_j}{\hat{p}_j} + \frac{1-t_j}{1-\hat{p}_j} \right) \frac{\partial \hat{p}_j}{\partial A} = 0; \tag{32}$$

$$\frac{\partial F(A,B)}{\partial B} = -\sum_j \left(\frac{t_j}{\hat{p}_j} + \frac{1-t_j}{1-\hat{p}_j} \right) \frac{\partial \hat{p}_j}{\partial B} = 0. \tag{33}$$

Differentiate both sides of (32) and (33) with respect to $\mathbf{v}^i$, we have

$$\begin{aligned} \frac{\partial^2 F(A,B)}{\partial \mathbf{v}^i \partial A} &= \sum_j \left(\frac{t_j}{\hat{p}_j^2} - \frac{1-t_j}{(1-\hat{p}_j)^2} \right) \frac{\partial p_j}{\partial A} \frac{\partial \hat{p}_j}{\partial \mathbf{v}^i} \\ &\quad - \sum_j \left(\frac{t_j}{\hat{p}_j} + \frac{1-t_j}{1-\hat{p}_j} \right) \left(\frac{\partial^2 \hat{p}_j}{\partial^2 A} \frac{\partial A}{\partial \mathbf{v}^i} + \frac{\partial^2 \hat{p}_j}{\partial B \partial A} \frac{\partial B}{\partial \mathbf{v}^i} + \frac{\partial^2 \hat{p}_j}{\partial f_j \partial A} \frac{\partial f_j}{\partial \mathbf{v}^i} \right) \\ &= 0; \end{aligned} \tag{34}$$

$$\begin{aligned} \frac{\partial^2 F(A,B)}{\partial \mathbf{v}^i \partial B} &= \sum_j \left(\frac{t_j}{\hat{p}_j^2} - \frac{1-t_j}{(1-\hat{p}_j)^2} \right) \frac{\partial p_j}{\partial B} \frac{\partial \hat{p}_j}{\partial \mathbf{v}^i} \\ &\quad - \sum_j \left(\frac{t_j}{\hat{p}_j} + \frac{1-t_j}{1-\hat{p}_j} \right) \left(\frac{\partial^2 \hat{p}_j}{\partial^2 B} \frac{\partial B}{\partial \mathbf{v}^i} + \frac{\partial^2 \hat{p}_j}{\partial A \partial B} \frac{\partial A}{\partial \mathbf{v}^i} + \frac{\partial^2 \hat{p}_j}{\partial f_j \partial B} \frac{\partial f_j}{\partial \mathbf{v}^i} \right) \\ &= 0. \end{aligned} \tag{35}$$

Note that $\partial \hat{p}_j/\partial \mathbf{v}^i$ of (34), (35) are further expressed in terms of $\partial A/\partial \mathbf{v}^i$ and $\partial B/\partial \mathbf{v}^i$ using (27), while $\partial f_j/\partial \mathbf{v}^i$ is known from (28), (31). Hence, $\partial A/\partial \mathbf{v}^i$and $\partial B/\partial \mathbf{v}^i$ can be solved from this expanded set of equations derived from (34–35).

The evaluation of $\partial \hat{p}_j/\partial \mathbf{v}^i$ involves the full set of training samples and is often computationally expensive. Fortunately, numerical evidence shows that the magnitudes of the 2nd and 3rd terms in the RHS of (27) are typically several orders smaller than the 1st term. Hence, an approximate value of $\partial \hat{p}_j/\partial \mathbf{v}^i$ can be found by making the assumption that $\partial A/\partial \mathbf{v}^i = 0$ and $\partial B/\partial \mathbf{v}^i = 0$. Under this assumption, $\partial \hat{p}_j/\partial \mathbf{v}^i$ reduces to the evaluation of the 1st term in the RHS of (27), which can be obtained by (28) and (31). Our numerical experiments use this approximation.

References

Boser, B., Guyon, I., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). San Mateo: Kaufmann.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *Proceedings of the 15th international conference on machine learning* (pp. 82–90). San Francisco: Kaufmann.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chang, C. C., & Lin, C. J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*, 131–159.

Chu, W., Keerthi, S. S., & Ong, C. J. (2003). Bayesian trigonometric support vector classifier. *Neural Computation*, *15*(9), 2227–2254.

Chu, W., Keerthi, S. S., & Ong, C. J. (2004). Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, *15*(1), 29–44.

Cortes, C., & Vapnik, V. N. (1995). Support vector networks. *Machine Learning*, *20*(3), 273–297.

Cristianini, N., & Shawe-Taylor, J. (2000). *Introduction to support vector machines*. Cambridge: Cambridge University Press.

Duan, K. B., & Keerthi, S. S. (2005). Which is the best multiclass SVM method? An empirical study. *Lecture Notes in Computer Science*, *3541*, 278–285.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. N. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422.

Guyon, I., et al. (2003). *NIPS 2003 feature selection competition*. http://www.nipsfsc.ecs.soton.ac.uk/.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2006a). *Feature extraction, foundations and applications*. Berlin: Springer.

Guyon, I., Gunn, S., Hur, A. B., & Dror, G. (2006b). Feature selection benchmark for classification problems. In I. Guyon, S. Gunn, M. Nikravesh, & L. A. Zadeh (Eds.), *Feature extraction, foundations and applications*. Berlin: Springer.

Günter, S., & Bunke, H. (2004). Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. *Pattern Recognition Letters*, *25*(11), 1323–1336.

Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *The Annals of Statistics*, *26*(2), 451–471.

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods: support vector learning*. Cambridge: MIT Press.

Keerthi, S. S. (2002). Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, *13*(5), 1225–1229.

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

Lal, T. N., Chapelle, O., Weston, J., & Elisseeff, A. (2006). Embedded methods. In I. Guyon, S. Gunn, M. Nikravesh, & L. A. Zadeh (Eds.), *Feature extraction, foundations and applications*. Berlin: Springer.

Lee, J. H., & Lin, C. J. (2000). *Automatic model selection for support vector machines* (Technical Report). Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. Online at http://www.csie.ntu.edu.tw/cjlin/papers/modelselect.ps.gz.

Lin, H. T., Lin, C. J., & Weng, R. C. (2003). *A note on Platt's probabilistic outputs for support vector machines* (Technical Report). Department of Computer Science, National Taiwan University. http://www.csie.ntu.edu.tw/~cjlin/papers/plattprob.ps.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., & Müller, K.-R. (1999). Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, & S. Douglas (Eds.), *Neural networks for signal processing IX* (pp. 41–48). New York: IEEE.

Neumann, J., Schnörr, C., & Steidl, G. (2005). Combined SVM-based feature selection and classification. *Machine Learning*, *61*(1–3), 129–150.

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Page, E. S. (1967). A note on generating random permutations. *Applied Statistics*, *16*(3), 273–274.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods: support vector learning*. Cambridge: MIT Press.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, & D. Schuurmans (Eds.), *Advances in large margin classifiers*. Cambridge: MIT Press.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, *3*, 1357–1370.

Rätsch, G. (2005). *Benchmark repository*. http://ida.first.fhg.de/projects/bench/benchmarks.htm.

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

Saon, G., & Padmanabhan, M. (2001). Minimum Bayes error feature selection for continuous speech recognition. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems* (Vol. 13, pp. 800–806).

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Vapnik, V. N., & Chapelle, O. (2000). Bounds on error expectation for support vector machines. *Neural Computation*, *12*(9), 2013–2036.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. N. (2001). Feature selection for SVMs. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems* (Vol. 13, pp. 668–674).

Williams, C. K. I., & Rasmussen, C. E. (1996). Gaussian processes for regression. In D.S. Touretzky, M.C. Mozer, & M.E. Hasselmo (Eds.), *Advances in neural information processing systems* (Vol. 8, pp. 598–604).