Expert Systems with Applications 38 (2011) 12506–12513

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Support Vector Machine incorporated with feature discrimination

Yunyun Wang [a], Songcan Chen [a,*], Hui Xue [b]

[a] Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, 210016 Nanjing, China
[b] School of Computer Science & Engineering, Southeast University, 210016 Nanjing, China

ARTICLE INFO

Keywords:
Weight vector
Feature (attribute) discrimination
Weight penalization matrix
Support Vector Machine
Pattern classification

ABSTRACT

Support Vector Machine (SVM) achieves state-of-the-art performance in many real applications. A guarantee of its performance superiority is from the maximization of between-class margin, or loosely speaking, full use of discriminative information from between-class samples. While in this paper, we focus on not only such discriminative information from samples but also discrimination of individual features and develop feature discrimination incorporated SVM (FDSVM). Instead of minimizing the l_2-norm of feature weight vector, or equivalently, imposing equal penalization on all weight components in SVM learning, FDSVM penalizes each weight by an amount decreasing with the corresponding feature discrimination measure, consequently features with better discrimination can be attached greater importance. Experiments on both toy and real UCI datasets demonstrate that FDSVM often achieves better performance with comparable efficiency.

1. Introduction

Support Vector Machine (SVM) first introduced by Vapnik (1998) and Cortes and Vapnik (1995) has been successfully applied to many applications, including face detection (Osuna, Freund, & Girosi, 1997), document categorization (Joachims, 1999), gene selection (Guyon, Weston, Barnhill et al., 2002) and financial forecast (Ding, Song, & Zen, 2008; Li & Sun, 2009). Its basic motivation is to find the separating hyper-plane with the maximum margin between classes (Burges, 1998; Cristianini & Taylor, 2000), as a result, SVM makes full use of the discriminative information between samples from different classes. Recently, several generalized formulations of SVM have successively been proposed with aim to make better use of the underlying prior information in the given samples and achieved improved performance. Belkin, Niyogi, and Sindhwani (2006) proposed LapSVM through incorporating the manifold structure information of data into SVM and brought significant performance promotion in semi-supervised learning. Xue, Chen, and Yang (2008) developed the Structure Regularized Support Vector Machine (SRSVM) through embedding the cluster-structure information into SVM, and boosted the classification performance distinctly by both maximizing the margin between classes and minimizing the compactness within individual classes. Decoste and Schölkopf (2002), Pozdnoukhov and Bengio (2006) and Schölkopf, Burges, and Vapnik (1996) developed different variants of SVM respectively by introducing the transformation invariance so that the classification is invariant to some transformations in the input space. Shivaswamy and Jebara (2006) developed the π-SVM through introducing the permutation invariance to ensure the invariance of the classifier to permutations of sub-elements in each input. Besides, Akbani, Kwek & Japkowicz (2004) and Wang and Japkowicz (2009) incorporated the unbalanced class-distribution information into SVM and boosted its performance by setting different trade-off parameters for different classes. A relative comprehensive review of such researches can be found in Lauer and Bloch (2008).

In general, the input samples are represented by vectors of features or attributes (Krupka & Tishby, 2007), then the given sample set can be represented either by 'a set of *samples*' (Akbani, Kwek & Japkowicz, 2004; Burges, 1998; Cristianini & Taylor, 2000; Decoste & Schölkopf, 2002; Schölkopf et al., 1996; Taylor & Cristianini, 2004) or by 'a set of *features* or *attributes*' (Fei, Quanz, & Huan, 2009; Krupka & Tishby, 2007; Sandler, Talukdar, & Ungar, 2008; Wang, 2008). In such a sense, SVM and all above-mentioned variants can be viewed as being built on the underlying information in the *sample space* (the space spanned by samples). While in this paper, we also focus on the information underlying in the *feature space* (the space spanned by features or attributes) and attempt to incorporate it into SVM learning for better classification. Up to date, there have already appeared some algorithms utilizing the prior information in the feature space. Tibshirani, Saunders, Rosset et al. (2005) considered some meaningful feature orders and proposed the "fused lasso", which penalizes the l_1-norm of both the feature weights and their successive differences, thus encourages both sparsity and local constancy for the weights. Li and Li (2008) incorporated the feature graphical structure represented by the Laplacian matrix into the linear regression model, and

* Corresponding author. Tel.: +86 25 84896481x12221; fax: +86 25 84892811.
E-mail address: s.chen@nuaa.edu.cn (S. Chen).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.034

proposed a network-constrained regularization procedure to impose the smoothness over the feature weights. Similarly, Fei et al. (2009) incorporated the feature graph Laplacian into Support Vector Machine and developed GLSVM to achieve the smoothness with respect to the reference feature network. Sandler et al. (2008) also considered the feature graph similarities in learning, and penalized each feature weight by the squared amount it differs from the average weights of its neighbors so that the weights can be smooth over the feature graph. Besides, Krupka and Tishby (2007) incorporated the feature-weight covariance matrix defined by the distances in a 'meta-feature' space into the SVM learning to ensure the feature weights being a smooth function of the meta-features.

Obviously, these typical algorithms are all developed from the *smoothness* assumption of the features, and the prior information represented by the feature graph (network) or the meta-features is independent of the given data, thus can be viewed as additional information out of the data. In this paper, different from the algorithms concentrating on the *discriminative information of the between-class samples in the sample space*, we also focus on the *discrimination of individual features in the feature space*, and different from the ones concerning more on the *smoothness* among features, we pay more attention on the discrimination of individual features, which can be derived directly from the given data, i.e., evaluated *by the data itself even though there is no prior information provided beforehand*. Such prior information of feature discrimination will not necessarily determine the final classification, but can provide helpful guidance for it. Consequently we develop a feature discrimination incorporated Support Vector Machine (FDSVM), in which the features with better discrimination are paid more attention as they usually manifest greater importance in separating data correctly (Ho & Basu, 2000; Ho & Basu, 2002; Wang, 2008). In the learning of FDSVM, the weights are respectively penalized by the amounts decreasing with their corresponding feature discrimination measures. Experiments on both toy and real UCI datasets (Blake & Merz, 1998) demonstrate that compared with SVM, FDSVM often achieves better generalization performance with comparable efficiency.

In addition, it is worth pointing out that such an incorporation of feature discrimination is general and can straightforwardly be applied to the above variants of SVM, and other regularized algorithms such as the least squared SVM, (Suykens & Vandewalle, 1999) for their further performance promotions.

The rest of the paper is organized as follows. Section 2 introduces the preliminary about the standard SVM and the measures adopted for evaluating the discrimination of individual features. Section 3 presents the formulation of the proposed FDSVM, along with its kernel extension and time complexity analysis. Section 4 shows the experiment results on both toy and real datasets and some conclusions are drawn in Section 5.

2. Preliminary

2.1. Support Vector Machine

Support Vector Machine (Burges, 1998; Cortes & Vapnik, 1995; Cristianini & Taylor, 2000; Vapnik, 1998) has been developed from the theory of statistical learning (Vapnik, 1998) and structural risk minimization, and achieved great success in pattern recognition (Ding et al., 2008; Guyon, Weston, Barnhill et al., 2002; Joachims, 1999; Li & Sun, 2009; Osuna et al., 1997). It separates the binary samples with the maximum sample-margin between classes (Burges, 1998; Cristianini & Taylor, 2000; Xue et al., 2008) to control the complexity of the model and bound the generalization error (Burges, 1998; Cristianini & Taylor, 2000; Lauer & Bloch, 2008). For the linear case, given the training set $S = \{(x_i, y_i)\}_{i=1}^n$ with $x_i \in R^d$ and $y_i \in \{-1, 1\}$, and a linear decision function $f(x) = w^T x + b$, the optimization problem of SVM can be formulated as

$$\begin{aligned} \min \quad & \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{1}$$

where ξ_is are the slack variables allowing samples to violate the constraints and C is the trade-off parameter controlling the compromise between the maximization of margin and accepted amount of violations (Lauer & Bloch, 2008). This problem leads to the following dual formulation through the standard Lagrange method,

$$\begin{aligned} \max \quad & \sum_{i=1}^{n}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & 0 \leqslant \alpha_i \leqslant C, i = 1, \ldots n \end{aligned} \tag{2}$$

where α_is are the Lagrange multipliers and those non-zeros α_is correspond to the support vectors lying in the margin or strictly on the margin boundaries. The resulting decision function can be described as $f(x) = sign\left(\sum_{i=1}^{n}\alpha_i y_i x_i^T x + b\right)$.

For the non-linear case, the input samples are first mapped to a higher, even infinite dimension kernel space where a linear separation is feasible, and then through the implementation of SVM in the kernel space, a non-linear separating hyper-plane in the original input space can be obtained with improved classification. However, due to the infinite dimension of the kernel space, the non-linear mapping function $\phi : \mathrm{R}^d \to \mathscr{H}$ can not be formulated explicitly. An effective solution is to express all computations in terms of dot products $\varphi(x_i)^T\varphi(x_j)$ and use a kernel function $K(x_i, x_j)$ to replace them, which is the so-called 'kernel trick' (Cristianini & Taylor, 2000; Taylor & Cristianini, 2004; Xue et al., 2008).

2.2. Measures of feature discrimination

Features with better discrimination have manifested greater importance in separating data correctly, thus are usually emphasized in feature selection (Fei et al., 2009; Ho & Basu, 2000, 2002; Li & Li, 2008; Tibshirani, Saunders, Rosset et al., 2005; Wang, 2008). The discrimination of individual features can be evaluated directly from the given data and there have been several such evaluation measures presented in literature (Ho & Basu, 2000, 2002; Wang, 2008). In this paper, we adopt three of them, i.e., the Fisher's discriminant ratio, the separated region ratio and the feature efficiency, they are all supervised measures and emphasize on the geometrical characteristics of the class distributions which is the most critical for classification accuracy (Ho & Basu, 2000, 2002). In what follows, they are introduced respectively in separate sub-sections.

2.2.1. Fisher's discriminant ratio (F_1)

The Fisher's discriminant ratio is a classic measure for supervised class separability, which is defined as the ratio of the squared between-class scatter and the within-class scatter,

$$F_1 = \frac{(m_1 - m_2)^2}{\delta_1^2 + \delta_2^2} \tag{3}$$

where m_1, m_2, δ_1^2, δ_2^2 are the means and variances of two classes respectively along the given feature (Ho & Basu, 2000, 2002).

2.2.2. Ratio of separated region (F_2)

The ratio of the separated region along a given feature f can be formulated as

$$F_2 = \frac{\max(\min(f^+), \min(f^-)) - \min(\max(f^+), \max(f^-))}{\max(\max(f^+), \max(f^-)) - \min(\min(f^+), \min(f^-))} \tag{4}$$

where $\max(f^+)$, $\max(f^-)$, $\min(f^+)$ and $\min(f^-)$ are the maximum and minimum feature values in individual classes respectively. Therefore after the normalization of the feature, F_2 is identical to the *margin* of the closest between-class samples if the given samples are linear separable, and the negative of the overlap-region length otherwise, obviously, a larger F_2 value indicates a better feature discrimination.

2.2.3. Feature efficiency (F_3)

The feature efficiency measures the proportion of samples not lying in the overlap region (Ho & Basu, 2000, 2002). Specifically, let $a = \max(\min(f^+), \min(f^-))$ and $b = \min(\max(f^+), \max(f^-))$, it can be formulated as

$$F_3 = \begin{cases} 1 - \frac{\#(f \mid f \in [a,b])}{n}, & \text{if } a \leqslant b, \\ 1, else, \end{cases} \tag{5}$$

where $\#(S)$ denotes the number of elements in S and n is the total number of samples in both classes. Thus if there is no overlap between classes, F_3 achieves the maximum value of 1, and F_3 is a lower bound of *accuracy* when the feature values are viewed as classification scores with the classification hyper-plane perpendicular to the feature.

As a result, F_1 focuses on the between-class scatter represented by the difference of the between-class sample means, and the within-class scatter simultaneously, F_2 emphasizes the minimum sample-margin between classes and F_3 more concerns the number of separated samples, thus bounding the accuracy. Of course, other feature discrimination measures can also be applied here, while here we just focus on the above three ones due to the commonality in such an incorporation manner.

3. Feature discrimination incorporated Support Vector Machine

Support Vector Machine delivers the state-of-art performance in many real applications through maximizing the margin between classes (Burges, 1998; Cristianini & Taylor, 2000; Xue et al., 2008), as a result, SVM concentrates much on the discriminative information between samples from different classes (Xue et al., 2008). The margin term in the objective function of SVM is formulated as the magnitude squares of the weight vector $\boldsymbol{w}$, i.e. $\|w\|^2$ or $\sum_{i=1}^{d} w_i^2$ where d is the dimension of the features, thereby SVM penalizes all weight components in $\boldsymbol{w}$ equally. In this paper, we focus on not only the discriminative information of samples but also the discrimination of individual features and propose the feature discrimination incorporated Support Vector Machine (FDSVM), in which each component in $\boldsymbol{w}$ is penalized by an amount decreasing with the corresponding feature discrimination measure. The formulation of the proposed FDSVM will be presented in the next subsection, followed by its kernel extension and time complexity analysis.

3.1. Formulation of FDSVM

We first define the feature discrimination vector $q = \{q_i\}_{d\times 1}$ with each q_i describing the discrimination of the ith feature. To ensure the (feature) weight penalization parameter positive and decreasing with the increase of q_i, we further define the weight penalization matrix $A = (a_{ij})_{d\times d}$ with each a_{ij} formulated as

$$a_{ij} = \begin{cases} \exp(-\eta q_i), & i = j \\ 0, & i \neq j \end{cases} \tag{6}$$

η is the parameter scaling the diagonal elements in A. A is diagonal and positive definite, with the diagonal elements decreasing with the increase of the corresponding feature discrimination measures and reflecting the penalization degrees of the corresponding feature weights. To avoid the possible "domination" of some features, each diagonal element in A is normalized through $a_{ii} = a_{ii}/trace(A)$, $i = 1 \ldots d$ so that $0 \leqslant a_{ii} \leqslant 1$ and $\sum_{i=1}^{d} a_{ii} = 1$. Finally the primal problem of FDSVM can be formulated as

$$\begin{aligned} \min \quad & \tfrac{1}{2} w^T A w + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0, i = 1, \ldots n \end{aligned} \tag{7}$$

or equivalently,

$$\begin{aligned} \min \quad & \tfrac{1}{2} \sum_{i=1}^{d} a_{ii} w_i^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0, i = 1, \ldots n \end{aligned} \tag{8}$$

From (8), it is clear that the weights of the better discrimination features are relatively less penalized through the embedding of A, thus those better discrimination features can be attached greater importance in learning. Further, when the value of η approaches 0, A is close to an identity matrix and hence the proposed FDSVM degenerates to the standard SVM. Through introducing the positive Lagrange multipliers for each inequality constraints in (7), or (8), the corresponding dual problem can be derived as

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T A^{-1} x_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \\ & 0 \leqslant \alpha_i \leqslant C, i = 1, \ldots n \end{aligned} \tag{9}$$

which is similar to the dual problem of the standard SVM in (2) but replacing $x_i^T x_j$ with $x_i^T A^{-1} x_j$, amounting to transforming S to a new one $\tilde{S}$ by $T: x \to A^{-1/2}x$, and the training of SVM on the new dataset $\tilde{S}$ is equivalent to the training of FDSVM on the original one S. The diagonal elements in $A^{-1/2}$ increase with the feature discrimination measures, thus the features with better discrimination are relatively emphasized after the transformation T. The finally-derived decision function can be written as

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i x_i^T A^{-1} x \tag{10}$$

and the specific learning algorithm of FDSVM is presented in the Table 1,

Table 1
The learning algorithm of FDSVM.

Input	$x_i \in R^d$, $i \in \{1 \ldots n\}$ – the ith input sample
	$y_i \in \{-1, 1\}$, $i \in \{1 \ldots n\}$ – the label of x_i
	C – the trade-off parameter
	η – the scaling parameter
Output	w – the feature weight vector, or projection norm-vector for classification
	b – the classification threshold
Procedure	Evaluate the feature discrimination vector $q = \{q_i\}_{d\times 1}$ using F_1 (or F_2, F_3)
	Construct the weight penalization matrix $A = (A_{ij})_{d\times d}$ with η and (6)
	Transform the original data through T: $x_i \to A^{-1/2}x_i$, $i \in \{1 \ldots n\}$
	Perform SVM on the transformed dataset with C, output (w, b)

3.2. The kernel extension of FDSVM

As described in Section 2.1, for many linear non-separable applications, the linear algorithms in the input space are often not so powerful due to their under-fitting, thus we usually implicitly map the samples from the input space to a higher dimension feature space in which a linear separation is feasible, and then perform the linear algorithms in such implicit feature space with the help of the popular 'kernel trick' (Cristianini & Taylor, 2000; Taylor & Cristianini, 2004; Xue et al., 2008). However, we fail to directly apply this 'kernel trick' used in SVM to the proposed FDSVMs, since they involve the discrimination measures evaluated *explicitly* on *individual* features, which are *implicitly expressed* in the feature space induced by kernel, thus we instead adopt an alternative kernelization approach in which the implicit mapping is replaced by the explicit empirical kernel mapping (EKM) (Scholkopf, Mika, & Burges, 1999; Xiong, Swamy, & Ahmad, 2005) such that the explicit evaluation of feature discrimination becomes possible. In the following, we will detail the EKM process through which to extend the linear FDSVM to its kernel version.

Now let $K_{train} = [k_{ij}]_{n\times n}$ with respect to the kernel matrix from the training set, due to its symmetrical positive-semidefinition, it can be decomposed as $K_{train} = U\,\Lambda\,U^T$, where $\Lambda \in R^{n\times n}$ is a diagonal matrix containing its Eigen-values and $U \in R^{n\times n}$ consists of the corresponding Eigen-vectors. Then the EKM for mapping samples from the input space to the empirical kernel space can be formulated as

$$x \to \Lambda^{-1/2}U^T(k(x,x_1),k(x,x_2)\ldots k(x,x_n))^T \tag{11}$$

thus the training and testing set in the empirical kernel space can be represented as $X^e_{train} = \Lambda^{-1/2}UK_{train}$ and $X^e_{test} = \Lambda^{-1/2}UK_{test}$ respectively. Further, if the kernel matrix K_{train} is singular or we want to reduce the dimension of the empirical kernel space, we can select the non-zero or top-r-rank Eigen-values and their corresponding Eigen-vectors, i.e. $K_{train} = U^r\,\Lambda^r U^{rT}$ with $U^r \in R^{n\times r}$ and $\Lambda^r \in R^{r\times r}$ $(r \leqslant n)$, then the new training and testing sets through the reduced EKM become $X^e_{train} = \Lambda^{r-1/2}U^rK_{train}$ and $X^e_{test} = \Lambda^{r-1/2}U^rK_{test}$ respectively. Finally the kernel version FDSVM can be derived through evaluating the individual feature discrimination and then performing the linear version FDSVM on such newly-derived data with the predefined kernel.

It is worthy noting that the discrimination vector here is evaluated from the n features in the empirical kernel space rather than the d features in the input space, since the original input samples have been non-linearly mapped to the empirical kernel space and the features on which the kernel version FDSVM really constructed are the ones in such empirical kernel space.

3.3. Time complexity analysis

Given the data $X \in R^{d\times n}$, where n is the number of samples and d is the number of features, the training time of the original SVM is $O(n^3)$ (Burges, 1998; Cristianini & Taylor, 2000). For the linear case, the time for calculating the feature discrimination measure F_1 (or F_2 and F_3) incorporated in FDSVM is $O(nd)$. Then from Table 1, the total training times of the proposed FDSVMs become $O(nd + d + 2nd + n^3)$, which is comparable to that of the original SVM except the case of $n^2 << d$, in which we can first use some feature extractor as a preprocessor for the given data to reduce its dimension (Fei et al., 2009; Wang, 2008). For the nonlinear case, the time for calculating F_1 (or F_2 and F_3) becomes $O(n^2)$, and the learning times of FDSVMs are $O(n^2 + n + 2n^2 + n^3)$, if only r $(r \leqslant n)$ Eigen-values and the corresponding Eigen-vectors are selected, the training times of FDSVMs become $O(nr + r + 2nr + n^3)$, thus still comparable to that of SVM. In conclusion, FDSVM can maintain comparable efficiency with the original SVM.

4. Experiment

In this section, we will evaluate the performance of the proposed FDSVM (FDSVM1, FDSVM2 and FDSVM3 corresponding to F_1, F_2 and F_3 respectively) by comparing with SVM on both toy and real datasets. In the toy problem, we use a two-dimension dataset containing two Gaussian distributions, in which the data projections along one dimension are more separable than those along the other, and compare FDSVM and SVM both with the linear kernel. In the real problem, we select 14 UCI datasets (Blake & Merz, 1998) to compare the performances of FDSVM and SVM, using both the linear and Gaussian kernels. We resort to the popular LIBSVM toolbox (Chang & Lin, 2001) in our experiment to learn both FDSVM and SVM, and search the regularization parameter C and the scaling parameter η from the set $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, and the width parameter σ of the Gaussian kernel from $\{2^{-10}, 2^{-9}, \ldots, 2^9, 2^{10}\}$ respectively through 5-fold cross-validation.

4.1. Toy problem

The two-dimension binary dataset used here is described in Table 2, each class follows a Gaussian distribution and the

Table 2
The attributes of the toy dataset.

CLASS	MEAN	COVARIANCE	NUMBER
CLASS 1	[0, 0]	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$	100
CLASS 2	[2, 1]		100

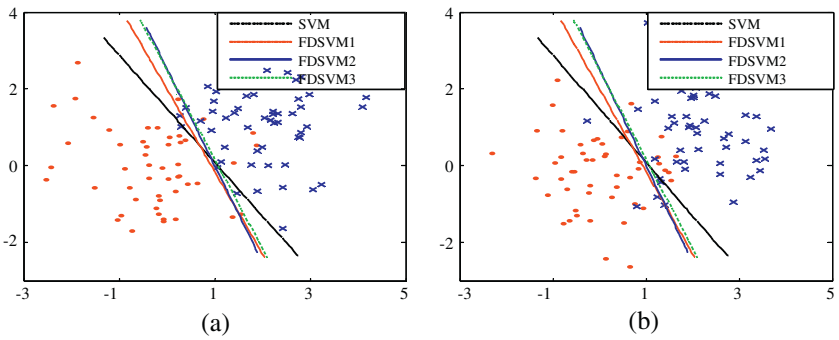

Fig. 1. The (a) training and (b) testing dataset and the corresponding separating hyper-planes derived from the linear kernel SVM and the linear kernel FDSVMs.

projections of data along the x axis are more separable than those along the y axis, since the difference of the class sample means along the x axis is larger than that along the y axis under the same class sample covariance. Each class has 100 samples, from which we randomly select a half as the training set, and the rest as the testing set, the distribution of the dataset is shown in Fig. 1, where '·' and '×' denote the samples in individual classes respectively.

The linear kernel FDSVMs and linear kernel SVM are respectively performed on the dataset, the resulting accuracies along with the testing times are listed in Table 3, and their corresponding separating hyper-planes are displayed in Fig. 1. Besides, the feature discrimination vectors and the weight penalization matrices in individual FDSVMs are shown in Table 4, and the performances of FDSVMs according to different values of η are illustrated in Fig. 2. Jointly from those tables and figures we can make the following observations,

- From Fig. 1, the separating hyper-planes of FDSVMs are all less inclined compared to that of SVM, i.e. more close to the separating hyper-plane derived exclusively from the x-feature (more discriminative feature according to the data generation), or the separating hyper-plane perpendicular to the x axis. Thus through such incorporations, better discrimination features can be attached greater importance and contribute more to the between-class separation.
- From Table 3, FDSVMs obtain the same training accuracy as SVM while achieve better accuracies on the testing set. Besides, the testing times of FDSVMs are all close to that of SVM, which

Table 3
The resulting accuracies and testing times of the linear kernel SVM and the linear kernel FDSVMs, the bold value indicates that FDSVM performs better than SVM, and value with underline indicates the best performance among those four classifiers.

CLASSIFIER	SVM	FDSVM1	FDSVM2	FDSVM3
TRAIN_ACC.	0.92	0.92	0.92	0.92
TEST_ACC.	0.85	**0.88**	**0.88**	**0.89**
TEST_TIME	0.125	0.1994	0.2031	0.1969

Table 4
The corresponding feature discrimination vectors and weight penalization matrices in individual FDSVMs.

CLASSIFIER	FDSVM1	FDSVM2	FDSVM3
q_1:q_2	2.8748:0.5359	−0.2384:−1.0368	0.7:0.01
a_{11}:a_{22}	0.4418:0.5582	0.3104:0.6896	0.334:0.666

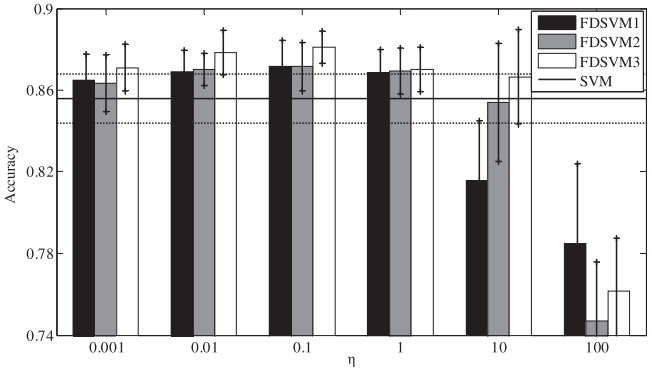

Fig. 2. The performances (testing accuracy and standard deviation) of the linear kernel SVM and the linear kernel FDSVMs with respect to different values of η with C fixed to 1.

Table 5
The average accuracies and variances from SVM and FDSVMs on 14 UCI datasets, using both the linear and the Gaussian kernels, the values with "∗" indicate significant performance improvements from FDSVMs through the t-test, and the values with underline indicate the best accuracies among SVM and FDSVMs with the linear and Gaussian kernels respectively.

DATASET	SVM		FDSVM1		FDSVM2		FDSVM3	
	LINEAR	GAUS.	LINEAR	GAUS.	LINEAR	GAUS.	LINEAR	GAUS.
ARRHYTHMIA	.6718	.7426	**.6923**∗	**.7613**∗	**.6981**∗	**.7613**∗	**.6871**	**.7666**∗
	.0009	.0011	.0008	.0007	.0011	.001	.0015	.0011
AUTOMOBILE	.8280	.8300	**.8532**∗	**.8671**∗	**.8524**∗	**.8644**∗	**.8516**∗	**.8662**∗
	.0016	.0019	.0022	.0012	.0025	.0012	.0025	.0012
BIOMED	.8526	.8750	.8514	**.8949**	**.8696**	**.8884**∗	**.8688**	**.8899**∗
	.0005	.0006	.0006	.0018	.0001	.0009	.0004	.0009
BUPA	.6368	.6926	.6279	.6886	.6432	**.7027**	**.6568**∗	**.7129**∗
	.0047	.0023	.0041	0018	.0047	.0021	.0038	.0019
ECHOCARDIOGRAM	.8484	.8645	**.8659**∗	**.8769**∗	.8449	.8531	**.8593**	**.8765**
	.001	.0016	.0017	.0009	.0013	.001	.0019	.0019
HEPATITIS	.6264	.7705	**.7139**∗	**.7827**∗	**.6723**∗	.7740	**.7234**∗	**.7896**∗
	.0109	.0014	.0033	.0012	.0067	.0009	.0044	.0007
HORSE_COLIC	.5254	.7087	**.6281**∗	**.7268**∗	**.5893**∗	**.7232**∗	**.5991**∗	**.7256**∗
	.0042	.0006	.0024	.0006	.0031	.001	.0038	.0008
HOUSE	.9014	.9234	**.9256**∗	**.9350**∗	**.9145**	**.9397**∗	**.9258**∗	**.937**∗
	.0007	.0004	.0006	.0008	.0007	.0002	.0006	.0002
HOUSING	.6254	.9263	**.6515**∗	**.9387**∗	**.6501**∗	**.9386**	**.6693**∗	**.9385**∗
	.0019	.0004	.0018	.0004	.0017	.0007	.0015	.0003
IMPORT	.8282	.8334	.8347	.8388	**.8532**∗	**.8624**∗	**.8487**	**.843**∗
	.0009	.0015	.0009	.0019	.0025	.0014	.0025	.0013
IONOSPHERE	.8450	.8907	.8453	**.9067**∗	.8537	.8950	.8510	.8993
	.0006	.0008	.0006	.0008	.0004	.0009	.0005	.0012
PIMA	.6747	.7448	.6761	.7514	**.7074**∗	**.7522**	**.6915**∗	**.7639**∗
	.0042	.0013	.0038	.0016	.0009	.0015	.001	.0012
SONAR	.7177	.8684	**.7535**∗	**.8882**∗	.7271	**.8834**∗	.7104	.8755
	.0013	.0025	.0009	.0022	.0016	.0017	.0024	.0014
WDBC	.8941	.9190	**.9065**	**.9431**∗	**.9214**∗	**.9378**∗	.9040	.9054
	.0016	.0008	.0009	.0004	.0003	.0006	.0018	.0008
MEAN	.7483	.8279	**.7733**	**.8429**	**.7712**	**.8412**	**.7748**	**.8422**
CASE (WIN/t-TEST)	14/14	14/14	**9/8**	**11/10**	**10/8**	**11/8**	**11/7**	**11/9**

is consistent to our analysis in Section 3.3. As a result, FDSVMs can achieve better generalization performance than SVM through the incorporation of feature discrimination while maintain comparable efficiency.

- From Table 4, the feature discrimination values (q1) with respect to the x-feature are all relatively larger than those (q2) of the y-feature, consequently the corresponding weight penalization parameters (a11) are relatively smaller and the weights (w1) are penalized with lower degrees. Meanwhile, the weight penalization parameters in individual FDSVMs are not exactly the same, since they rely on the initial feature discrimination vectors, whereas we can still adjust them through the choose of the scaling parameter, as a result, the feature discrimination incorporated can help but not determine the final classification.
- In Fig. 2, the most suitable values of η are all in the range [0.01,1], further, when η approaches to 0, the performances of FDSVMs would be close to that of SVM, and when it becomes too large, the performances of FDSVMs would decrease, as the final classification would be dominated completely by the x-feature, or the feature with better discrimination.

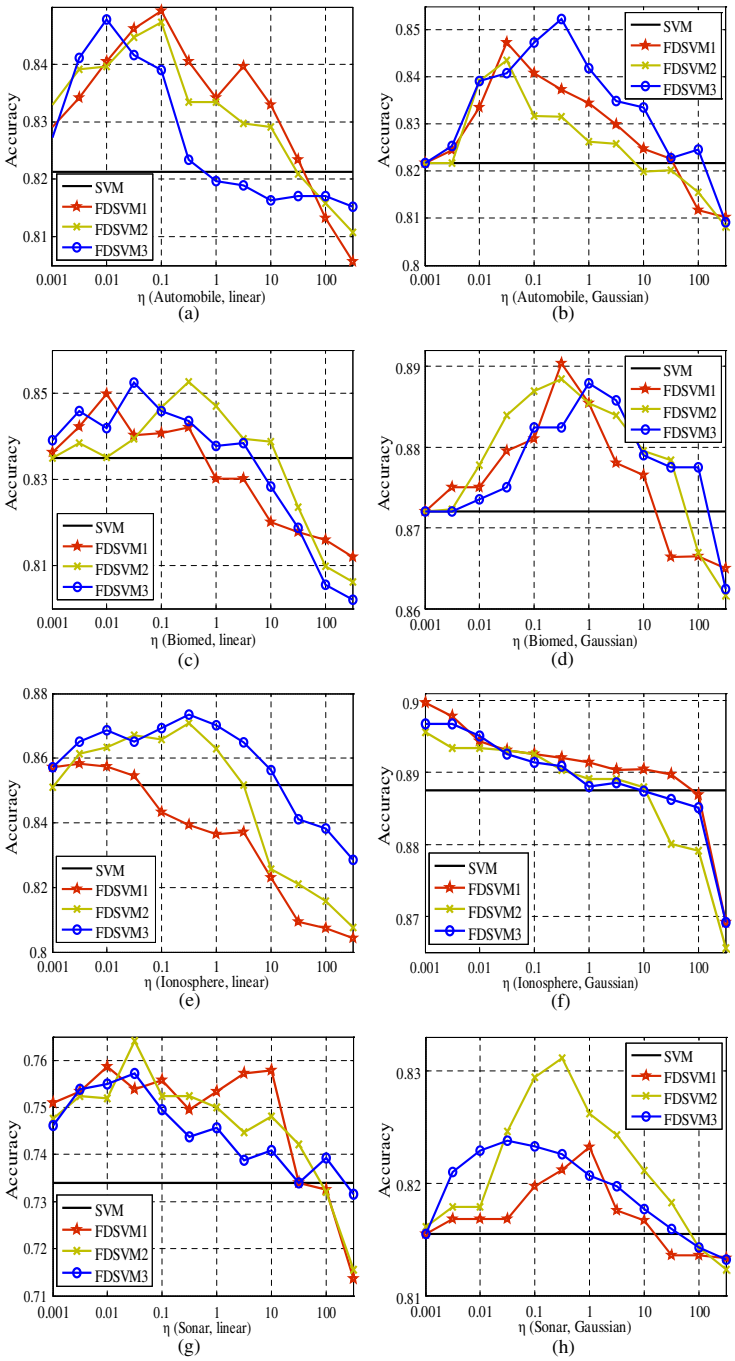

Fig. 3. The performances of SVM and FDSVMs with respect to η on 4 selected UCI datasets, *automobile* with the (a) linear and (b) Gaussian kernel, *biomed* with the (c) linear and (d) Gaussian kernel, *ionosphere* with the (e) linear and (f) Gaussian kernel and *sonar* with the (g) linear and (h) Gaussian kernel, for the linear kernel, $C = 1$, and for the Gaussian kernel, $C = 1$, $\sigma = 1$ ($\sigma = 64$ for the biomed dataset).

4.2. Real problems

In this subsection we select 14 UCI datasets to compare the performances of SVM and FDSVMs, using both the linear and Gaussian kernels. For each dataset, we randomly select a half of the samples in each class for training and the rest for testing, this process for each algorithm is repeated 10 times and the average results are reported in Table 5. Each row (except the last two ones) in the table gives the average testing accuracies and variances of the individual algorithms on each dataset, the bold values in each row indicate that FDSVMs outperform SVM by at least one percent in performance, the values with '*' indicate significant performance improvements from FDSVMs through the *t*-test with the confidence interval at 95%, and the values with underline indicate the best accuracies among SVM and FDSVMs with the linear and the Gaussian kernels respectively. The last second row gives the average testing accuracies of individual algorithms on the overall datasets and the last row shows the number of cases in which FDSVMs achieve performance improvement by at least one percent and significant improvement through the t-test compared with SVM. Then we can conclude that

- When the linear kernel is used, FDSVM1 outperforms SVM on 9 out of the 14 given datasets and has significant improvement on 8 datasets through the t-test, FDSVM2 achieves better performance on 10 datasets with significant improvement on 8 ones, and FDSVM3 performs better on 11 datasets with significant improvement on 7 ones, when the Gaussian kernel is used, FDSVM1 outperforms SVM on 11 datasets with significant improvement on 10 ones, FDSVM2 performs better than SVM with significant improvement on 9 ones, and FDSVM3 outperforms SVM on 11 datasets and achieves significant improvement on 8 ones. Moreover, the average accuracies on all datasets from FDSVMs are all larger than that from the original SVM, with both the linear and the Gaussian kernels. As a result, FDSVMs can achieve better generalization performance than SVM through the incorporation of feature discrimination.
- Table 5 also lists the datasets on which FDSVMs perform no better than SVM, the main reason can be that the discrimination evaluations of individual features are based on the training samples, thus the discriminative information derived respectively from the sample and the feature spaces is consistent to some extent. SVM has made full use of the discriminative information in the sample space, therefore partially utilized the discriminative information hidden in the feature space implicitly, as a result, the explicit incorporation of feature discrimination may lead to no distinct performance-promotion when the discriminative information in the feature space incorporated has already been utilized in SVM implicitly, and boosted performance otherwise.
- As shown in Table 5, different feature discrimination measures lead to different performance promotions from SVM, thus stating that the selection of proper feature discrimination measure is application-oriented, but it is not our focus in this paper and deserves another exploration.
- Fig. 3 reveals the performances of SVM and FDSVMs according to different values of η on four selected datasets, from which we can see that when η approaches 0, the accuracies of FDSVMs become close to those of SVM, since the (feature) weight penalization matrices are all close to an identity matrix, and when η becomes larger, the accuracies tend to increase to some maximal values, then gradually decrease and become even lower than those of SVM, which can attribute to the exclusive dominations of some better discrimination features. It is not suitable for the ionosphere dataset with the Gaussian kernel, the reason is that we simply set C to 1 and the range for η is not large enough for revealing such rule.

5. Conclusions

In this paper we propose a novel feature discrimination incorporated Support Vector Machine (FDSVM) through utilizing not only the discriminative information from between-class samples, which is emphasized in the original SVM, but also the discrimination of individual features, i.e., discriminative prior information from both the *sample* and the *feature* spaces. During the learning of FDSVM, the feature weight components in $\boldsymbol{w}$ are respectively penalized according to their individual feature discrimination, and the better discrimination features can obtain more attention and contribute more to the between-class classification, as they usually manifest greater importance in separating data correctly. Compared with SVM, FDSVMs are empirically validated to be able to achieve better generalization performance in most cases while maintain comparable efficiency.

It is worth pointing out that the proposed incorporation of feature discrimination is general and can straightforwardly be applied to other variants of SVM or other regularized algorithms for their further performance promotions. Moreover, any feature discrimination measures or feature importance values provided beforehand can conveniently be used to develop FDSVM targeting to specific applications. As a result, we will attempt to incorporate prior feature importance combining specific applications in our feature work, and meanwhile, search for the feature importance in the optimization process to find out whether it can achieve better performance at the price of efficiency.

Acknowledgments

This work is partially supported by Natural Science Foundations of China Grant Nos. 60973097, 61035003 and 60905002.

References

Akbani, R., Kwek, S. & Japkowicz, N. 2004. Applying Support Vector Machines to Imbalanced Datasets. In *Proceedings of the 15th European Conference on Machine Learning* Vol. 3201, pp. 39–50.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research, 7*(12), 2399–2434.

Blake, C. L., Merz, C. J. 1998. *UCI repository of machine learning databases*.

Burges, C. J. C. (1998). A tutorial on support vector machine for pattern recognition. *Data Mining Knowledge Discovery, 2*(2), 121–167.

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.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning, 20*(3), 273–279.

Cristianini, N., & Taylor, J. S. (2000). *An introduction to Support vector Machines and other kernel-based learning methods*. UK: Cambridge University Press.

Decoste, D., & Schölkopf, B. (2002). Training invariant Support vector machines. *Machine Learning, 46*(1–3), 116–190.

Ding, Y., Song, X., & Zen, Y. (2008). Forecasting financial condition of Chinese listed companies based on support vector machine. *Expert Systems with Applications, 34*(4), 3081–3089.

Fei, H., Quanz, B., & Huan, J. 2009. GLSVM: Integrating structured feature selection and large margin classification. In *ICDM 2009 Workshop on Optimization Based Methods for Emerging Data Mining Problems*.

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

Ho, T. K., & Basu, M. 2000. Measuring the complexity of classification problems. In *Proceedings of the 15th International Conference on Pattern Recognition*, Vol. 15, pp. 37–43.

Ho, T. K., & Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 289–300.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of international conference machine learning* pp. 200–209.

Krupka, E., & Tishby, N. 2007. Incorporating prior knowledge on features into learning. *In Proceedings of 11th International Conference on Artificial Intelligence and Statistics*.

Lauer, F., & Bloch, G. (2008). *Incorporating prior knowledge in support vector machines for classification: A review Neurocomputing, 71*(7-9), 1578–1594.

Li, C., & Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics, 24*(9), 1175–1182.

Li, H., & Sun, J. (2009). Predicting business failure using multiple case-based reasoning combined with support vector machine. *Expert Systems with Applications, 36*(6), 10085–10096.

Osuna, E., Freund, R., & Girosi, F. 1997. Training support vector machines: Application to face detection. In *Proceedings of Computer Vision and Pattern Recognition* pp. 130–136).

Pozdnoukhov, A., & Bengio, S. (2006). Invariances in kernel methods: from samples to objects. *Pattern Recognition Letters, 27*(10), 1087–1097.

Sandler, T., Talukdar, P. P., & Ungar, L. H. 2008. Regularized Learning with Networks of Features. In *the 21st Advance in Neural Information Processing Systems* pp. 1401–1408.

Schölkopf, B., Burges, C., & Vapnik, V. 1996. Incorporating invariances in support vector learning machines. In *Proceedings of the 1996 International Conference on Artificial Neural Networks* pp. 47–52.

Scholkopf, B., Mika, S., & Burges, C. J. C. (1999). Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks, 10*(5), 1000–1017.

Shivaswamy, P., & Jebara, T. 2006. Permutation invariant SVMs. In *Proceedings of the 23rd International Conference on Machine Learning* pp. 817–824.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*(3), 293–300.

Taylor, J. S., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. UK: Cambridge University Press.

Tibshirani, R., Saunders, M., Rosset, S., et al. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B, 67*(1), 91–108.

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

Wang, B. X., & Japkowicz, N. (2009). Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems, 19*(2), 1–20.

Wang, L. (2008). Feature selection with kernel class separability. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(9), 1534–1546.

Xiong, H., Swamy, M. N. S., & Ahmad, M. O. (2005). Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks, 16*(2), 460–474.

Xue, H., Chen, S., & Yang, Q. 2008. Structural Support Vector Machine. In *Proceedings of the 15th International Symposium on Neural Networks* pp. 505–511.