Structural Support Vector Machine

Hui Xue[1], Songcan Chen[1,⋆], and Qiang Yang[2]

[1] Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, 210016, Nanjing, P.R. China
[2] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
{xuehui,s.chen}@nuaa.edu.cn, qyang@cse.ust.hk
http://parnec.nuaa.edu.cn

Abstract. Support Vector Machine (SVM) is one of the most popular classifiers in pattern recognition, which aims to find a hyperplane that can separate two classes of samples with the maximal margin. As a result, traditional SVM usually more focuses on the scatter between classes, but neglects the different data distributions within classes which are also vital for an optimal classifier in different real-world problems. Recently, using as much structure information hidden in a given dataset as possible to help improve the generalization ability of a classifier has yielded a class of effective large margin classifiers, typically as Structured Large Margin Machine (SLMM). SLMM is generally derived by optimizing a corresponding objective function using SOCP, and thus in contrast to SVM developed from optimizing a QP problem, it, though more effective in classification performance, has the following shortcomings: 1) large time complexity; 2) lack of sparsity of solution, and 3) poor scalability to the size of the dataset. In this paper, still following the above line of the research, we develop a novel algorithm, termed as Structural Support Vector Machine (SSVM), by directly embedding the structural information into the SVM objective function rather than using as the constraints into SLMM, in this way, we achieve: 1) to overcome the above three shortcomings; 2) empirically better than or comparable generalization to SLMM, and 3) theoretically and empirically better generalization than SVM.

Keywords: Support vector machine, Structural information, Rademacher complexity, Pattern recognition.

1 Introduction

In the past decade, large margin machines have become a hot issue of research in machine learning. Support Vector Machine (SVM)[1], as the most famous one among them, is derived from statistical learning theory[2] and achieves a great success in pattern recognition.

⋆ Corresponding author: Tel: +86-25-84896481 Ext.12106; Fax: +86-25-84498069. This work was supported respectively by NSFC (60773061) and Jiangsu NSF (BK2008xxx).

F. Sun et al. (Eds.): ISNN 2008, Part I, LNCS 5263, pp. 501–511, 2008.

Given a training set $\{\mathbf{x}_i, y_i\}_{i=1}^n \in \mathbf{R}^m \times \{\pm 1\}$, the basic objective of SVM is to learn a classifier $f = \mathbf{w}^T\mathbf{x} + b$ which can maximize the margin between classes:

$$\min_{\mathbf{w},b} \ \frac{1}{2} \parallel \mathbf{w} \parallel^2$$

$$s.t. \ \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \ \ i = 1, \cdots, n \tag{1}$$

If we focus on the constraints in (1), we can immediately capture the following insight about SVM which is easily generalized to the soft margin version:

Theorem 1. SVM constrains the scatter between classes as $\mathbf{w}^T\mathbf{S}_b\mathbf{w} \geq 4$, where $\mathbf{S}_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$, μ_i is the mean of class $i (i = 1, 2)$.

Proof. Without loss of generalization, we assume that the class one has the class label $y_i = 1$, and the other class has $y_j = -1$. Then we reformulate the constraints as $\mathbf{w}^T\mathbf{x}_i + b \geq 1$, where $\mathbf{x}_i$ belongs to class one, and $\mathbf{w}^T\mathbf{x}_j + b \leq -1$, where $\mathbf{x}_j$ belongs to class two. Let the numbers of the samples in the two classes are respectively n_1 and n_2. Then we have $\frac{1}{n_1}\sum_{i=1}^{n_1}(\mathbf{w}^T\mathbf{x}_i + b) = (\mathbf{w}^T\mu_1 + b) \geq 1$ and $-\frac{1}{n_2}\sum_{j=1}^{n_2}(\mathbf{w}^T\mathbf{x}_j + b) = -(\mathbf{w}^T\mu_2 + b) \geq 1$. Adding the two inequalities, we obtain $\mathbf{w}^T(\mu_1 - \mu_2) \geq 2$. Squaring the inequality, we further have $\mathbf{w}^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\mathbf{w} \geq 4$, i.e. $\mathbf{w}^T\mathbf{S}_b\mathbf{w} \geq 4$. ■

Consequently, following the above theorem, it is obvious that SVM actually gives a natural lower bound to the scatter between classes, just according with its original motivation that pays more attention to the maximization of margin. However, it discards the prior data distribution information within classes which is also vital for classification. In fact, corresponding to different real-world problems, different classes may have different underlying data structures. It requires that the classifier should adjust the discriminant boundaries to fit the structures which are vital for classification, especially for the generalization capacity of the classifier. However, the traditional SVM does not differentiate the structures, and the derived decision hyperplane lies unbiasedly right in the middle of the support vectors[3, 4], which may lead to a nonoptimal classifier in the real-world problems.

Recently, some new large margin machines have been presented to give more concerns to the structural information than SVM. They provide a novel view to design a classifier, that the classifier should be sensitive to the structure of the data distribution, and assume that the data contains clusters. Minimax Probability Machine (MPM)[5] and Maxi-Min Margin Machine (M^4)[3] stress the global structure of the two classes and apply two ellipsoids, i.e. two clusters, to characterize the classes distributions respectively. By using the Mahalanobis distance which combines the mean and covariance of the ellipsoids, they integrate the global structural information into the large margin machines. However, only emphasis on the global structure of the classes is too coarse. In many real-world problems, samples within classes more likely have different distributions. Therefore, Structured Large Margin Machine (SLMM)[4] is proposed to firstly

apply some clustering methods to capture the underlying structures in each class. As a result, SLMM uses several ellipsoids whose number is equal to the number of the clusters to enclose the training data, rather than only two ellipsoids in M^4 respectively corresponding to each class. The optimization problem in soft margin SLMM can be formulated as (2)[4], which introduces the covariance matrices in each cluster into the constraints:

$$\max \ \rho - C \sum_{l=1}^{|P|+|N|} \xi_l$$

$$s.t. \ (\mathbf{w}^T \mathbf{x}_l + b) \geq \frac{\mid P_i \mid}{Max_P} \rho \sqrt{\mathbf{w}^T \Sigma_{P_i} \mathbf{w}} - \xi_l, \ \mathbf{x}_l \in P_i,$$

$$-(\mathbf{w}^T \mathbf{x}_l + b) \geq \frac{\mid N_j \mid}{Max_N} \rho \sqrt{\mathbf{w}^T \Sigma_{N_j} \mathbf{w}} - \xi_l, \ \mathbf{x}_l \in N_j,$$

$$\mathbf{w}^T \mathbf{r} = 1, \ \xi_l \geq 0 \tag{2}$$

where ξ_l is the penalty for violating the constraints. C is a regularization parameter that makes a trade-off between the margin and the penalties incurred. P_i denotes the ith cluster in class one, $i = 1, \cdots, C_P$, and N_j denotes the jth cluster in class two, $j = 1, \cdots, C_N$. C_P and C_N are the numbers of the clusters in the two classes respectively. $\mathbf{r}$ is a constant vector to limit the scale of the weight $\mathbf{w}$.

By the simple algebraic deduction, MPM, M^4 even SVM can all be viewed as the special cases of SLMM. And SLMM also achieves better classification performance among these popular large margin machines experimentally. However, SLMM has much larger time complexity than SVM. Its optimization problems should be solved by SOCP, which handles relatively difficultly in real applications. And the corresponding solution loses the sparsity as in SVM derived from optimizing a QP problem. Consequently, it has poor scalability to the size of the dataset and can not easily be generalized to large-scale or multi-class problems. Furthermore, in the kernel version, SLMM should kernelize the covariance matrix in each cluster within the constraints respectively, which undoubtedly increases extra computational complexity.

In this paper, we present a novel classification algorithm that provides a general way to incorporate the structural information into the learning framework of the traditional SVM. We call our method SSVM, which stands for Structural Support Vector Machine. Inspired by the SLMM, SSVM also firstly exploits the intrinsic structures of samples within classes by some unsupervised clustering methods, but then directly introduces the data distributions of the clusters in different classes into the traditional optimization function of SVM rather than in the constraints. The contributions of SSVM can be described as follows:

- SSVM naturally integrates the prior structural information within classes into SVM, without destroying the classical framework of SVM. And the corresponding optimization problem can be solved by the QP just similarly to SVM. Consequently, SSVM can overcome the above shortcomings of SLMM.

- SSVM empirically has comparable or better generalization to SLMM, since it considers the separability between classes and the compactness within classes simultaneously. Though SLMM can capture the structural information within classes by some clustering algorithms, it also more emphasizes the separability between classes due to the characteristics of the traditional large margin machines, which more likely does not sufficiently apply the prior information to some extent.
- SSVM can be theoretically proved that it has the lower Rademacher complexity than SVM, in the sense that it has better generalization capacity, rather than only validating generalization performance empirically in SLMM. This further justifies that the introduction of the data distribution within classes into the classifier design is essential for better recognition.

The rest of the paper is organized as follows. Section 2 presents the proposed Structural Support Vector Machine, and also discusses the kernelization of SSVM. In Section 3, the theoretical analysis of the generalization capacity is deduced. Section 4 gives the experimental results. Some conclusions are drawn in Section 5.

2 Structural Support Vector Machine (SSVM)

Following the line of the research in the SLMM, SSVM also has two steps: clustering and learning. It firstly adopts clustering techniques to capture the data distribution within classes, and then minimizes the compactness in each cluster, which leads to further maximizing the margin in the sense of incorporating the data structures simultaneously.

Many clustering methods, such as K-means, nearest neighbor clustering and fuzzy clustering, can be applied in the first clustering step. After the clustering, the structural information is introduced into the objective function by the covariance matrices of the clusters. So the clusters should be compact and spherical for the computation. Following SLMM, here we use the Ward's linkage clustering in SSVM, which is one of the hierarchical clustering techniques. During the clustering, the Ward's linkage between clusters to be merged increases as the number of clusters decreases[4]. We can draw a curve to represent this process. Through finding the knee point, i.e. the point of maximum curvature in the curve, the number of clusters can be determined automatically. Furthermore, the Ward's linkage clustering is also applicable in the kernel space.

After clustering, we obtain the c_1 and c_2 clusters respectively in the two classes. We denote the clusters in the classes as $P_1, \cdots, P_{c_1}$ and $N_1, \cdots, N_{c_2}$. From Theorem 1, we have proved that SVM gives a natural lower bound to the separability between classes by the constraints. So here we pay more attention to the compactness within classes, that is, the clusters which cover the different structural information in different classes. We aim to maximize the margin and simultaneously minimize the compactness. Accordingly, the SSVM model in the soft margin version can be formulated as:

$$\min_{\mathbf{w},b} \ \frac{1}{2} \parallel \mathbf{w} \parallel^2 + \frac{\lambda}{2} \mathbf{w}^T \Sigma \mathbf{w} + C \sum_{i=1}^{n} \xi_i$$

$$s.t.\ \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,\ \ \xi_i \geq 0,\ \ i = 1, \cdots, n \tag{3}$$

where $\Sigma = \Sigma_{P_1} + \cdots + \Sigma_{P_{c_1}} + \Sigma_{N_1} + \cdots + \Sigma_{N_{c_2}}$, Σ_{P_i} and Σ_{N_j} are the covariance matrices corresponding to the ith and jth clusters in the two classes, $i = 1, \cdots, c_1$, $j = 1, \cdots, c_2$. λ is the parameter that regulates the relative importance of the structural information within the clusters, $\lambda \geq 0$.

Compared to SVM, SSVM inherits the advantages of SLMM that incorporates the data distribution information in a local way, that considers the covariance matrices of the clusters in each class which contain the trend of data occurrence in statistics[4]. However, different from SLMM, SSVM directly introduces the prior information into the objective function rather than the constraints. Therefore, SSVM can follow the same techniques as SVM to solve the optimization problem, which mitigates the large computational complexity in SLMM. And the algorithm can efficiently converge to the global optimum which also holds the sparsity and has better scalability to the size of the datasets. Moreover, through minimizing the compactness of the clusters, SSVM more likely further maximizes the margin between classes, which may lead to comparable or better classification and generalization performance than SLMM. We will address these in more details in the following sections.

By incorporating the constraints into the objective function, we can rewrite (3) as a primal Lagrangian. Then, we transform the primal into the dual problem following the same steps as SVM:

$$\max_{\alpha}\ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j[\mathbf{x}_i^T(\mathbf{I} + \lambda\Sigma)^{-1}\mathbf{x}_j]$$

$$s.t.\ \ 0 \leq \alpha_i \leq C,\ \ i = 1, \cdots, n$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{4}$$

Eq. (4) is a typical convex optimization problem. Using the same QP techniques as SVM, we can obtain the solution . Then the derived classifier function can be formulated as follows, which is used to predict the class labels for future unseen samples $\mathbf{x}$:

$$f(\mathbf{x}) = sgn[\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i^T(\mathbf{I} + \lambda\Sigma)^{-1}\mathbf{x} + b] \tag{5}$$

It is noteworthy that SSVM boils down to the same solution framework of SVM except adding a regularization parameter λ. When $\lambda = 0$, SSVM will degenerate to the traditional SVM. Thus SVM actually can be viewed as a special version case of SSVM.

We can also apply the kernel trick in SSVM in order to further improve the classification performance in complex pattern recognition problems. Furthermore, compared to SLMM which has to kernelize each cluster covariance matrix respectively, SSVM can perform complex kernelization through kernelizing the

covariance matrix sum of all the cluster covariance matrices which makes it simpler and more effective.

Assume that the nonlinear mapping function is $\Phi : \mathbf{R}^m \rightarrow \mathbf{H}$, where $\mathbf{H}$ is a Hilbert space which has high dimension. Then the optimization function of SSVM in the kernel space can be described as:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j [\Phi(\mathbf{x}_i)^T (\mathbf{I} + \lambda \Sigma^{\Phi})^{-1} \Phi(\mathbf{x}_j)]$$

$$s.t. \;\; 0 \leq \alpha_i \leq C, \;\; i = 1, \cdots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{6}$$

Due to the high dimension (even infinite), Φ usually can not be explicitly formulated. A solution to this problem is to express all computations in terms of dot products, called as the kernel trick[1]. The kernel function $k : \mathbf{R}^m \times \mathbf{R}^m \rightarrow \mathbf{R}$, $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ derives the corresponding kernel matrix $\mathbf{K} \in \mathbf{R}^{n \times n}$, $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, which is so-called Gram matrix.

Consequently, we aim to transform (6) into the form of dot products for adopting the kernel trick. For each covariance matrix in the kernel space, we have

$$\Sigma_i^{\Phi} = \frac{1}{\mid C_i^{\Phi} \mid} \sum_{\Phi(\mathbf{x}_j) \in C_i^{\Phi}} [\Phi(\mathbf{x}_j) - \mu_i^{\Phi}][\Phi(\mathbf{x}_j) - \mu_i^{\Phi}]^T$$

$$= \frac{1}{\mid C_i^{\Phi} \mid} \mathbf{T}_i^{\Phi} \mathbf{T}_i^{\Phi^T} - \mathbf{T}_i^{\Phi} \vec{\mathbf{1}}_{|C_i^{\Phi}|} \vec{\mathbf{1}}_{|C_i^{\Phi}|}^{T} \mathbf{T}_i^{\Phi^T} \tag{7}$$

where C_i^{Φ} denotes the clusters without differentiating the different classes, $i \in [1, c_1 + c_2]$. And $\mathbf{T}_i^{\Phi}$ is a subset of the sample matrix, which is combined with the samples belonging to the cluster i in the kernel space. $\vec{\mathbf{1}}_{|C_i^{\Phi}|}$ denotes a $\mid C_i^{\Phi} \mid$-dimensional vector with all the components equal to $1/ \mid C_i^{\Phi} \mid$.

Then we obtain

$$\Sigma^{\Phi} = \sum_{i=1}^{c_1+c_2} \Sigma_i^{\Phi} = \sum_{i=1}^{c_1+c_2} \frac{1}{\mid C_i^{\Phi} \mid} \mathbf{T}_i^{\Phi} \mathbf{T}_i^{\Phi^T} - \mathbf{T}_i^{\Phi} \vec{\mathbf{1}}_{|C_i^{\Phi}|} \vec{\mathbf{1}}_{|C_i^{\Phi}|}^{T} \mathbf{T}_i^{\Phi^T} \triangleq \mathbf{P}^{\Phi} \Psi \mathbf{P}^{\Phi^T} \tag{8}$$

where $\mathbf{P}^{\Phi} = [\mathbf{T}_1^{\Phi}, \cdots, \mathbf{T}_{c_1+c_2}^{\Phi}]$,

$$\Psi = \begin{pmatrix} \frac{1}{|C_1^{\Phi}|} \mathbf{I}_{|C_1^{\Phi}|} - \vec{\mathbf{1}}_{|C_1^{\Phi}|} \vec{\mathbf{1}}_{|C_1^{\Phi}|}^{T} & & \\ & \ddots & \\ & & \frac{1}{|C_{c_1+c_2}^{\Phi}|} \mathbf{I}_{|C_{c_1+c_2}^{\Phi}|} - \vec{\mathbf{1}}_{|C_{c_1+c_2}^{\Phi}|} \vec{\mathbf{1}}_{|C_{c_1+c_2}^{\Phi}|}^{T} \end{pmatrix}$$

and $\mathbf{I}_{|C_i^{\Phi}|}$ is a $\mid C_i^{\Phi} \mid \times \mid C_i^{\Phi} \mid$ identity matrix, $i \in [1, c_1 + c_2]$.

By the Woodbury's formula

$$(\mathbf{A}+\mathbf{UBV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{UB}(\mathbf{B}+\mathbf{BVA}^{-1}\mathbf{UB})^{-1}\mathbf{BVA}^{-1} \tag{9}$$

So

$$(\mathbf{I}+\lambda\Sigma^{\Phi})^{-1} = (\mathbf{I}+\lambda\mathbf{P}^{\Phi}\Psi\mathbf{P}^{\Phi^T})^{-1} = \mathbf{I}-\lambda\mathbf{P}^{\Phi}\Psi(\Psi+\lambda\Psi\mathbf{P}^{\Phi^T}\mathbf{P}^{\Phi}\Psi)^{-1}\Psi\mathbf{P}^{\Phi^T} \tag{10}$$

By substituting (10) into the optimization function (6), we have the kernel form of the dual problem (6) as follows:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j[\mathbf{K}_{ij} - \lambda\tilde{\mathbf{K}}_i^T\Psi(\Psi+\lambda\Psi\hat{\mathbf{K}}\Psi)^{-1}\Psi\tilde{\mathbf{K}}_j]$$

$$s.t.\ \ 0 \le \alpha_i \le C,\ i = 1,\cdots,n$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{11}$$

where $\tilde{\mathbf{K}}_i$ represents the ith column in the kernel Gram matrix $\tilde{\mathbf{K}}$, $\tilde{\mathbf{K}}_{ij} = k(\mathbf{x}_i^{C_t}, \mathbf{x}_j)$, $\mathbf{x}_i^{C_t}$ is the sample that is realigned corresponding to the sequence of the clusters, $t = 1,\cdots,c_1+c_2$. And $\hat{\mathbf{K}}$ is the kernel Gram matrix, $\hat{\mathbf{K}}_{ij} = k(\mathbf{x}_i^{C_t}, \mathbf{x}_j^{C_t})$.

3 Rademacher Complexity

In this section, we will discuss the generalization capacity of SSVM in theory. Different from SLMM which only validates its better generalization performance than SVM empirically by experiments, we will indeed prove that the introduction of the structural information within classes can improve the generalization bound compared to SVM. Here we adopt the Rademacher complexity measure[6] and show the new error bound is tighter.

In the traditional kernel machines, we are accustomed to using VC-dimension [2] to estimate the generalization error bound of a classifier. However, the bound involves a fixed complexity penalty which does not depend on the training data, thus can not be universally effective[6]. Recently, Rademacher complexity, as an alternative notion, is presented to evaluate the complexity of a classifier instead of the classical VC-dimension[7]. And for the kernel machines, we can obtain an upper bound to the Rademacher complexity:

Theorem 2 [6]**.** If $k : \mathbf{X}\times\mathbf{X} \to \mathbf{R}$ is a kernel, and $S = \{\mathbf{x}_1,\cdots,\mathbf{x}_n\}$ is a sample of points from $\mathbf{X}$, then the empirical Rademacher complexity of the classifier $\mathbf{F}_B$ satisfies

$$\hat{R}_n(\mathbf{F}_B) \le \frac{2B}{n}\sqrt{\sum_{i=1}^{n}k(\mathbf{x}_i,\mathbf{x}_j)} = \frac{2B}{n}\sqrt{tr(\mathbf{K})} \tag{12}$$

where B is the bound of the weights $\mathbf{w}$ in the classifier.

Following Theorem 2, we then give the complexity analysis of SSVM compared to SVM.

Theorem 3 (***Complexity Analysis***)**.** The upper bound of the empirical Rademacher complexity $\hat{R}_{SSVM}(f)$ in SSVM is at most the upper bound of $\hat{R}_{SVM}(f)$ in SVM, that is, $tr(K_{SSVM}) \leq tr(K_{SVM})$.

Due to limited space, here we omit the proof. The theorem states that there is an advantage to considering the separability between classes and the compactness within classes simultaneously, i.e. the structural information within the clusters, to further reduce the Rademacher complexity of the classifiers being considered. Intuitively, the minimization of the compactness in the clusters more likely leads to the larger margin compared to SVM, which means better generalization performance in practice. Theorem 3 just provides us a theoretical interpretation for the intuition.

4 Experiments

To evaluate the proposed Structural Support Vector Machine (SSVM) algorithm, we investigate its classification accuracies and computational efficiencies in several real-world UCI datasets. Since Structured Large Margin Machine (SLMM)[4] has been shown to be more effective than many relatively modern learning machines, such as Minimax Probability Machine (MPM)[5], Maxi-Min Margin Machine (M^4)[3] and Radial Basis Function Networks (RBFN) in terms of classification accuracies, in this experiment we just compare SSVM with SLMM and SVM. For each dataset, we divide the samples into two non-overlapping training and testing sets, and each set contains almost half of samples in each class respectively. This process is repeated ten times to generate ten independent runs for each dataset and then the average results are reported.

Due to the relatively better performance of the kernel version, here we uniformly compare the algorithms in the kernel and soft margin cases. The width parameter σ in the Gaussian kernel, and the regularization parameters C and λ are selected from the set $\{2^{-10}, 2^{-9}, \cdots, 2^{9}, 2^{10}\}$ by cross-validation. We apply Sequential Minimal Optimization (SMO) algorithm to solve the QP problems in SSVM and SVM, and SeDuMi program to solve the SOCP problem in SLMM.

The experimental results are listed in Table 1. In each block in the table, the first row is the training accuracy and variance. The second row denotes the testing accuracy and variance. And the third one is the average training time in the ten runs after the selection of the parameters. We can make several interesting observations from these results:

- SSVM is consistently superior to SVM in the overall datasets both in the training and testing accuracies, owing to the proper consideration of data distribution information. Furthermore, SSVM also outperforms SLMM in almost all the datasets except in Pima, because that SSVM simultaneously captures the separability between classes and the compactness within classes

Table 1. The training and testing accuracies (%), variances and average training time (sec.) compared between SSVM and SLMM, SVM in the UCI datasets

	SSVM	SLMM	SVM
Automobile	***96.25*** $\pm$ 0.01	95.31* $\pm$0.01	95.63* $\pm$0.01
	91.14 $\pm$ 0.00	88.63* $\pm$0.03	88.48* $\pm$0.01
	0.44	3.20	0.36
Bupa	***77.36*** $\pm$ 0.10	76.03* $\pm$0.15	75.68* $\pm$0.08
	76.18 $\pm$ 0.04	73.52* $\pm$0.12	73.06* $\pm$0.06
	1.23	18.77	0.89
Hepatitis	***84.10*** $\pm$ 0.01	82.59* $\pm$0.01	79.87* $\pm$0.00
	83.25 $\pm$ 0.00	79.82* $\pm$0.03	79.61* $\pm$0.01
	0.58	3.75	0.42
Ionosphere	***98.46*** $\pm$ 0.00	96.97* $\pm$0.03	96.80* $\pm$0.02
	97.52 $\pm$ 0.01	95.63* $\pm$0.05	95.11* $\pm$0.02
	1.17	5.71	0.79
Pima	79.65 $\pm$0.02	***80.63*** $\pm$ 0.05	76.04* $\pm$0.01
	78.63 $\pm$0.01	***79.46*** $\pm$ 0.02	77.08* $\pm$0.02
	12.53	72.14	7.67
Sonar	***95.58*** $\pm$ 0.02	95.27 $\pm$0.01	86.54* $\pm$0.15
	87.60 $\pm$ 0.07	86.21* $\pm$0.11	85.00* $\pm$0.13
	0.61	3.34	0.50
Water	***98.81*** $\pm$ 0.02	95.61* $\pm$0.10	98.47 $\pm$0.02
	98.69 $\pm$ 0.01	95.49* $\pm$0.12	90.51* $\pm$0.09
	0.39	1.56	0.29
Wdbc	***95.96*** $\pm$ 0.00	94.89* $\pm$0.05	92.54* $\pm$0.01
	95.72 $\pm$ 0.00	94.57* $\pm$0.03	94.25* $\pm$0.01
	3.58	43.65	2.77

$'*'$ Denotes that the difference between SSVM and the other two methods is significant at 5% significance level, i.e., t-value > 1.7341.

rather than only emphasizing the separability in SLMM which may miss some useful classification information. And the gap of the classification accuracies between the two algorithms in Pima is less than one percent.

- The training and testing accuracies of SSVM are basically comparable in the datasets, which further provides us an experimental validation for better generalization capacity than SVM, according with the theoretical analysis in Theorem 3. And the variances show the good stability of the SSVM algorithm.
- We also report the average training time of the three algorithms. SSVM is slower than SVM due to the clustering pre-processing. However, it is much quicker than SLMM, which adopts the SOCP as the optimizor rather than the QP in the SSVM. Consequently, in view of the efficiency as well as classification performance, SSVM is more likely the best option among the three algorithms.
- In order to find out whether SSVM is significantly better than SLMM and SVM, we perform the t-test on the classification results of the ten runs to calculate the statistical significance of SSVM. The null hypothesis H_0

demonstrates that there is no significant difference between the mean number of patterns correctly classified by SSVM and the other two methods. If the hypothesis H_0 of each dataset is rejected at the 5% significance level, i.e., the t-test value is more than 1.7341, the corresponding results in Table 1 will be denoted $'*'$. Consequently, as shown in Table 1, it can be clearly found that SSVM possesses significantly superior classification performance compared with the other two methods in almost all datasets, especially according to the testing accuracies. And in Pima, there seems to be no significant difference between SSVM and SLMM, i.e. t-value < 1.7341. This just accords with our conclusions.

5 Conclusion

In this paper, we propose a novel large margin machine called as Structural Support Vector Machine (SSVM). Following the research of SLMM, SSVM also firstly captures the data distribution information in the classes by some clustering strategies. Due to the insights about the constraints in the traditional SVM, we further introduce the compactness within classes according to the structural information into the learning framework of SVM. The new optimization problem can be solved following the same QP as SVM, rather than the SOCP in the recent related algorithms such as MPM, M^4 and SLMM. Consequently, SSVM not only has much lower time complexity but also holds the sparsity of the solution. Furthermore, we validate that SSVM has better generalization capacity than SVM both in theory and practice. And it also has better than or comparable classification performance to these related algorithms.

Throughout the paper, we discuss SSVM in the binary classification problems. However, SSVM can be easily generalized to the multi-class problems by using the vector labeled outputs techniques, and to large-scale problems through combining with the techniques of minimum enclosing ball[8]. These issues will be our future research.

References

1. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
2. Vapnik, V.: Statistical Learning Theory. Wiley, Chichester (1998)
3. Huang, K., Yang, H., King, I., Lyu, M.R.: Learning Large Margin Classifiers Locally and Globally. In: ICML (2004)
4. Yeung, D.S., Wang, D., Ng, W.W.Y., Tsang, E.C.C., Zhao, X.: Structured Large Margin Machines: Sensitive to Data Distributions. Machine Learning 68, 171–200 (2007)
5. Lanckriet, G.R.G., Ghaoui, L.E., Bhattacharyya, C., Jordan, M.I.: A Robust Minimax Approach to Classfication. JMLR 3, 555–582 (2002)

6. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
7. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. JMLR 3, 463–482 (2002)
8. Tsang, I.W., Kocsor, A., Kwok, J.T.: Simpler Core Vector Machines with Enclosing Balls. In: ICML (2007)