Structural Regularized Support Vector Machine: A Framework for Structural Large Margin Classifier

Hui Xue, Songcan Chen, and Qiang Yang, *Fellow, IEEE*

***Abstract*— Support vector machine (SVM), as one of the most popular classifiers, aims to find a hyperplane that can separate two classes of data with maximal margin. SVM classifiers are focused on achieving more separation between classes than exploiting the structures in the training data within classes. However, the structural information, as an implicit prior knowledge, has recently been found to be vital for designing a good classifier in different real-world problems. Accordingly, using as much prior structural information in data as possible to help improve the generalization ability of a classifier has yielded a class of effective structural large margin classifiers, such as the structured large margin machine (SLMM) and the Laplacian support vector machine (LapSVM). In this paper, we unify these classifiers into a common framework from the concept of "structural granularity" and the formulation for optimization problems. We exploit the quadratic programming (QP) and second-order cone programming (SOCP) methods, and derive a novel large margin classifier, we call the new classifier the structural regularized support vector machine (SRSVM). Unlike both SLMM at the cross of the cluster granularity and SOCP and LapSVM at the cross of the point granularity and QP, SRSVM is located at the cross of the cluster granularity and QP and thus follows the same optimization formulation as LapSVM to overcome large computational complexity and non-sparse solution in SLMM. In addition, it integrates the compactness within classes with the separability between classes simultaneously. Furthermore, it is possible to derive generalization bounds for these algorithms by using eigenvalue analysis of the kernel matrices. Experimental results demonstrate that SRSVM is often superior in classification and generalization performances to the state-of-the-art algorithms in the framework, both with the same and different structural granularities.**

***Index Terms*— Generalization bound, machine learning, structural granularity, support vector machine.**

Manuscript received July 21, 2010; revised October 13, 2010 and January 9, 2011; accepted January 9, 2011. Date of publication March 7, 2011; date of current version April 6, 2011. This work was supported in part by the National Natural Science Foundation of China under Grant 60773061, Grant 60973097, Grant 60905002, and Grant 61035003, the Natural Science Foundation of Jiangsu Province of China under Grant BK2008381, and the Hong Kong Competitive Earmarked Research Grant under Project N_HKUST624/09.

H. Xue is with the School of Computer Science and Engineering, Southeast University, Nanjing 210016, China (e-mail: hxue@seu.edu.cn).

S. Chen is with the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China. He is also with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210016, China (e-mail: s.chen@nuaa.edu.cn).

Q. Yang is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: qyang@cse.ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2108315

I. Introduction

In THE past decade, large margin classifiers have become a hot topic of research in machine learning. Support vector machine (SVM) [1], [2], as the most famous one among them, is derived from statistical learning theory [3] and achieves a great success in pattern recognition. The basic motivation of SVM is to find a hyperplane that can separate two classes of data with maximal margin [4]. However, SVM usually pays more attention to the separation between classes than the prior structural information within classes in data. In fact, for different real-world problems, different classes may have different underlying data structures. It is thus desirable that a classifier be adaptable to the discriminant boundaries to fit the structures in the data, especially for increasing the generalization capacities 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 [4]–[6], which may lead to nonoptimal classification results for future problems.

Recently, some algorithms have been developed to give more weightage to the structural information than SVM. They provide a novel view in which to design a classifier, that is, a classifier should be sensitive to the structure of the data distribution [5]. These algorithms are mainly divided into two kinds of approaches. The first one is *manifold assumption-based*, which assumes that the data actually lie on a submanifold in the input space. A typical paradigm in this approach is Laplacian support vector machine (LapSVM) [7]. LapSVM constructs a Laplacian graph for each class on top of the local neighborhood of each datum to form the corresponding Laplacian (matrix) to reflect the manifold structure of individual-class data. They are then embedded into the traditional framework of SVM as additional manifold regularization terms, where the latter is solved via quadratic programming (QP).

A second approach is *cluster assumption-based* [8], which assumes that the data contains clusters and deduces several popular large margin classifiers, such as ellipsoidal kernel machine (EKM) [9], minimax probability machine (MPM) [10], maxi-min margin machine (M^4) [4], and structured large margin machine (SLMM) [5]. EKM deems the whole data as a single global cluster and estimates the minimum volume bounding ellipsoid surrounding the data by using semidefinite programming (SDP), and then applies the estimated centroid

and covariance matrix of the ellipsoid to remap the data to a unit sphere where it is formulated as a SVM and thus solved by QP. It was proved in [9] that using such ellipsoid can get lower Vapnik–Chervonenkis dimension [3] than the usual bounding sphere in SVM, or, equivalently, implying better generalization capacity. Meanwhile, MPM and M^4 stress the different class structure in the data and utilize one single ellipsoid, as a cluster, to characterize each class (distribution) respectively in the binary classification. By using the class-related Mahalanobis distance, which combines the centroids (or means) and covariance matrices of the ellipsoids instead of the class-unrelated Euclidean distance to measure the distance between the data and the discriminant boundary, MPM and M^4 integrate the class structural information into the large margin classifier optimization problems as the new constraints. However, just using a single ellipsoid (cluster) to describe each class is generally too coarse. In fact, in many real-world problems, data that are within classes are more likely to have different (cluster) structures. This observation motivates us to extend SLMM further. We note that SLMM focuses on the underlying structures in each class and applies some unsupervised clustering techniques to capture such finer structural information. Consequently, SLMM uses multi-ellipsoids or multi-clusters to enclose the data of each class to characterize each class finer. The subsequent optimization problem in soft margin SLMM can be formulated as in [5] in terms of (1), which embeds the covariance matrices in each cluster into the constraints

$$\begin{aligned} &\max \rho - C \sum_{l=1}^{|P|+|N|} \xi_l \\ &\text{s.t. } (\boldsymbol{w}^T \boldsymbol{x}_l + b) \geq \tfrac{|P_i|}{Max_P} \rho \sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma}_{P_i} \boldsymbol{w}} - \xi_l, \quad \boldsymbol{x}_l \in P_i, \\ &-(\boldsymbol{w}^T \boldsymbol{x}_l + b) \geq \tfrac{|N_j|}{Max_N} \rho \sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma}_{N_j} \boldsymbol{w}} - \xi_l, \quad \boldsymbol{x}_l \in N_j, \\ &\boldsymbol{w}^T \boldsymbol{r} = 1, \qquad \xi_l \geq 0 \end{aligned} \tag{1}$$

where P_i denotes the ith cluster in class one, $i = 1, \ldots, C_P$, and N_j denotes the jth cluster in class two, $j = 1, \ldots, C_N$. C_P and C_N are the number of the clusters in the two classes respectively. $\boldsymbol{r}$ is a constant vector to limit the scale of the weight $\boldsymbol{w}$.

By simple algebraic deductions, MPM, M^4, and even SVM can all be viewed as the special cases of SLMM [5]. It can achieve better classification performance among these algorithms experimentally. However, since its optimization problem can only be formulated as a second-order cone programming (SOCP) rather than QP as in SVM, in contrast to SVM, SLMM not only needs much higher computational cost but also loses the sparsity of solution. Especially, in order to get the kernel version of SLMM, the covariance matrix in each cluster within the constraints has to be kernelized respectively, which undoubtedly increases extra computational complexity.

In this paper, we first introduce the concept of "structural granularity" to characterize the different data structures used in the design process of classifiers. Based on the different granularities and the formulations for optimization problems, we construct a common framework for these structural large margin classifiers, which provides us with a new perspective to categorize the existing classifiers and analyze new ones. Through a systematic analysis on the framework, we further derive a novel large margin classifier called structural regularized support vector machine (SRSVM), which stands for SRSVM. In this framework, SRSVM has the same cluster granularity as SLMM and likewise aims to exploit the intrinsic cluster structures in data within classes. However, different from SLMM, SRSVM naturally integrates the distributions of the clusters within different classes into the traditional optimization problem of SVM rather than in the constraints, which can be solved by QP rather than SOCP in SLMM. That is, SRSVM can follow the same optimization formulation as LapSVM which is with the (datum) point granularity to overcome high computational complexity and the non-spare solutions in SLMM. Furthermore, SRSVM embeds the within-class compactness and the between-class separability simultaneously into the optimization problem, rather than only emphasizing only one of the two aspects, respectively, in SVM and SLMM. In order to evaluate the generalization performances of these classifiers comprehensively, we also discuss their generalization bounds by using the eigenvalue analysis of the kernel matrices [11]–[13]. Comparisons both on the experimental and theoretical analyses are made to validate the superiority of our SRSVM to the other algorithms in the proposed framework.

The rest of this paper is organized as follows. Section II, introduces the structural granularity and constructs the framework. Section III, presents the proposed SRSVM, including the linear and nonlinear versions. Experimental results both on the toy and real-world problems are given in Section IV. In Section V, the theoretical analysis of the generalization bounds for the algorithms is deduced. Some conclusions are drawn in Section VI.

II. Framework for Structural Large Margin Classifier

After decades of in-depth study on SVM, researchers have proposed many improved algorithms to modify its performance [14]–[19], where structural algorithms are one of the popular research trends in recent years. In this section, we analyze SVM from the structural view and introduce the concept of "structural granularity," which allows us to derive a common framework for the recent SVM-based structural large margin classifiers.

A. SVM

For binary classification problems, given a training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n \in \boldsymbol{R}^m \times \{\pm 1\}$, the objective of SVM is to learn a classifier $f = \boldsymbol{w}^T \boldsymbol{x} + b$ that can maximize the margin between classes

$$\begin{aligned} &\min_{\boldsymbol{w},b} \tfrac{\|\boldsymbol{w}\|^2}{2} \\ &\text{s.t. } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \qquad i = 1, \ldots, n. \end{aligned} \tag{2}$$

The above formulation can be further relaxed to solve linearly nonseparable problems [9]

$$\begin{aligned} &\min_{\boldsymbol{w},b} \tfrac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{n} \xi_i \\ &\text{s.t. } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \qquad i = 1 \ldots, n \end{aligned} \tag{3}$$

where ξ_i is the penalty for violating the constraints. C is a regularization parameter that makes a tradeoff between the margin and the penalties incurred.

If we focus on the constraints in (2), we can immediately capture the following insight about SVM, which is easily generalized to the relaxation version.

Proposition 1: SVM constrains the separation between classes as $\boldsymbol{w}^T \mathbf{S}_b \boldsymbol{w} \geq 4$, where $\boldsymbol{S}_b = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$, $\boldsymbol{\mu}_i$ 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: $\boldsymbol{w}^T \boldsymbol{x}_i + b \geq 1$, where $\boldsymbol{x}_i$ belongs to class one; $\boldsymbol{w}^T \boldsymbol{x}_j + b \leq -1$, where $\boldsymbol{x}_j$ belongs to class two.

Let the numbers of the samples in the two classes be respectively n_1 and n_2. Then we have

$$1/n_1 \sum_{i=1}^{n_1} \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) = \left(\boldsymbol{w}^T \boldsymbol{\mu}_1 + b \right) \geq 1 \tag{4}$$

$$-1/n_2 \sum_{j=1}^{n_2} \left(\boldsymbol{w}^T \boldsymbol{x}_j + b \right) = -\left(\boldsymbol{w}^T \boldsymbol{\mu}_2 + b \right) \geq 1. \tag{5}$$

Adding the two inequalities (4) and (5), we obtain

$$\boldsymbol{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \geq 2. \tag{6}$$

Squaring the inequality (6), we further have

$$\boldsymbol{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{w} \geq 4. \tag{7}$$

That is, $\boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w} \geq 4$. ■

Consequently, following the above proposition, it is clear that SVM gives a natural lower bound for the separation between classes, exactly according to its original motivation that pays more attention to the maximization of margin. However, it more likely neglects the prior data structural information within classes, which is also vital for classification. A linear classifier example is illustrated in Fig. 1, where '*' and '·' denote the two classes, respectively. Here each class is generated via a mixture of two Gaussian distributions that have approximately perpendicular trends of data occurrence. As we mentioned before, SVM does not sufficiently utilize the structurally obvious information, and the derived decision plane, denoted by the dash line in Fig. 1(a), approximately lies in the middle of three support vectors [4]–[6] in the training set, which leads to inaccurate classification in the testing set [Fig. 1(b)]. However, a more reasonable decision plane should be as denoted by the solid line in Fig. 1. This boundary has almost parallel orientation to the '·' class data trend, and, at the same time, relatively far from the '*' class due to the approximately vertical trend of the corresponding data. Consequently, SRSVM has better classification performance both in the training and testing sets.

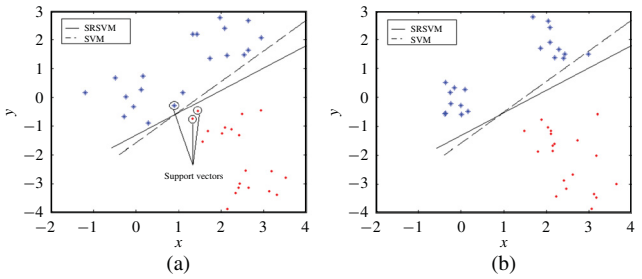

Fig. 1. Illustration on the importance of the structural information within classes in SRSVM and SVM. (a) Discriminant boundaries in the training set. (b) Discriminant boundaries in the testing set.

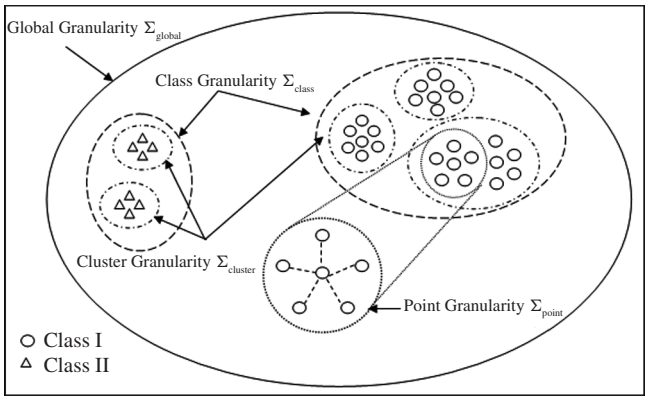

Fig. 2. Illustration of structural granularity.

B. Structural Granularity

Definition 1: Given a dataset $T = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$. Let $\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \mathbf{S}_t$ be a partition of $\boldsymbol{T}$ according to some relation measure, where the partition characterizes the whole data in the form of some structures such as cluster, and $\boldsymbol{S}_1 \cup \boldsymbol{S}_2 \cup \cdots \cup \boldsymbol{S}_t = \boldsymbol{T}$. Here $\boldsymbol{S}_i$ $(i = 1, 2, \ldots, t)$ is called structural granularity.

Clearly, structural granularity relies on the different assumptions about the actual data structures in real-world problems. In our viewpoint, it involves four layers, as illustrated in Fig. 2, where "○" and "△" denote the two classes respectively. Moreover, the data in the class I "○" are generated by three Gaussian distributions and the class II "△" are obtained by two Gaussian distributions.

According to the Gaussian mixture model [20] for a mixture Gaussian distributions, we can characterize the structural granularity of the training data by ellipsoids (or clusters), whose centroids (or means) and covariance matrices reflect the properties of Gaussian distributions. As a result, four granularity layers can be differentiated:

Global Granularity: The granularity refers to the dataset $\boldsymbol{T}$. With this granularity, the whole data are characterized or enclosed by a single ellipsoid, as shown by the solid line ellipsoid in Fig. 2, whose centroid $\boldsymbol{\mu}_{global}$ and covariance matrix $\boldsymbol{\Sigma}_{global}$ can be obtained by minimizing the volume of the ellipsoid [9]

$$\begin{aligned} &\min_{\boldsymbol{\Sigma}_{global}, \boldsymbol{\mu}_{global}} \ln \left| \boldsymbol{\Sigma}_{global} \right| \\ &\text{s.t.} \left\| \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{global} \right)^T \boldsymbol{\Sigma}_{global}^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{global} \right) \right\| \leq 1, \\ &\qquad \boldsymbol{\Sigma}_{global} \geq 0. \end{aligned} \tag{8}$$

The corresponding classifier, such as EKM, aims to utilize such global data structure, or more precisely, global data scatter in its design.

Class Granularity: The granularities are the class-partitioned data subsets. Single ellipsoid can be used to describe an individual class to form the so called class structure, as denoted by the dashed line ellipsoids in Fig. 2, whose mean and covariance matrix are defined by the data in the individual class [4]

$$\boldsymbol{\mu}^i_{class} = \frac{1}{n_i}\sum_{j=1}^{n_i}\boldsymbol{x}_j$$

$$\boldsymbol{\Sigma}^i_{class} = \frac{1}{n_i}\sum_{j=1}^{n_i}\left(\boldsymbol{x}_j - \boldsymbol{\mu}^i_{class}\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}^i_{class}\right)^T, \quad i = 1, 2. \quad (9)$$

Thus the corresponding classifiers, for example MPM and M^4, can focus on the global class scatter.

Cluster Granularity: The granularities are the data subsets within each class. The data structures within each class are depicted by a certain amount ellipsoids that are obtained by some clustering techniques, as shown by the dot and dash line ellipsoids in Fig. 2. The corresponding mean and covariance matrix in cluster i are [5]

$$\boldsymbol{\mu}^i_{cluster} = \frac{1}{n_{C_i}}\sum_{j=1}^{n_{C_i}}\boldsymbol{x}_j$$

$$\boldsymbol{\Sigma}^i_{cluster} = \frac{1}{n_{C_i}}\sum_{j=1}^{n_{C_i}}\left(\boldsymbol{x}_j - \boldsymbol{\mu}^i_{cluster}\right)\left(\boldsymbol{x}_j - \boldsymbol{\mu}^i_{cluster}\right)^T. \quad (10)$$

Compared to the above two kinds of classifiers, the cluster-granularity classifiers, including SLMM, have finer cluster assumption about the data.

Point Granularity: The granularities are the neighborhoods $ne(\boldsymbol{x}_i)$ of every datum $\boldsymbol{x}_i$, which are described by overlapped local ellipsoids surrounding the data in each class, as denoted by the dot line ellipsoid in Fig. 2, whose covariance matrix can be viewed as a kind of local generalized covariance [7]

$$\boldsymbol{\Sigma}^i_{point} = \sum_{\boldsymbol{x}_j \in ne(\boldsymbol{x}_i)} S_{ij}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \quad (11)$$

where

$$S_{ij} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}\right) & if\ \boldsymbol{x}_i \in ne(\boldsymbol{x}_j)\ or\ \boldsymbol{x}_j \in ne(\boldsymbol{x}_i) \\ 0 & otherwise. \end{cases}$$

The classifier with such granularity, as LapSVM, seeks the k nearest neighbors of individual samples within the same class to construct the local nearest neighbor scatter matrices $\boldsymbol{\Sigma}^i_{point}$. It then sums those matrices to form the global scatter matrices as Laplacians, which can characterize the data manifold structures in the respective classes.

C. Structural Large Margin Classifier Framework

Structural granularity reflects the data distribution arrangement from macroscopic to microcosmic, which offers a natural rule to reclassify the algorithms mentioned above. Consequently, we construct a new structural large margin classifier framework both from the structural granularity and formulation of optimization problem perspectives, as illustrated in Fig. 3.

Here we list two main optimization formulations in large margin classifiers, namely SOCP and QP, which denote the SOCP and QP methods, respectively. Each classifier is located at the cross of some granularity and optimization formulation. The "?" symbol represents that, to the best of our knowledge, the research on the corresponding classifier is still a blank.

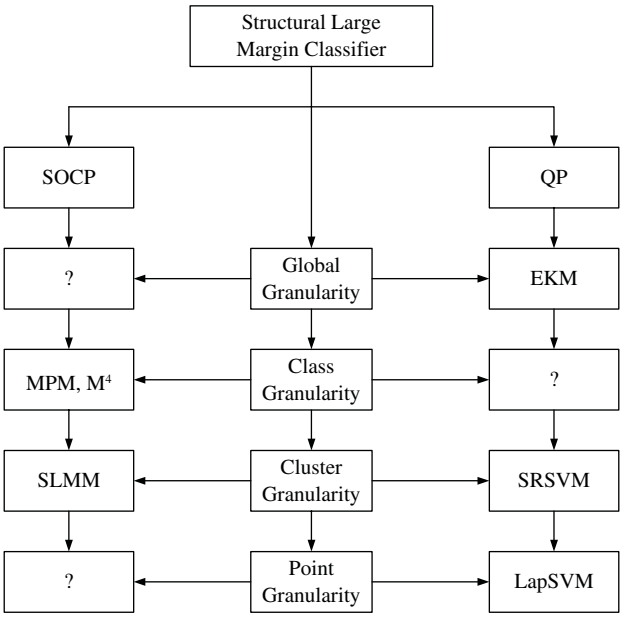

Fig. 3. Framework for structural large margin classifier.

From the vertical directions in the framework, due to the introduction of the structural information into the constraints, the optimization problems of MPM, M^4, and SLMM boil down to SOCP, and are more difficult to solve in real applications. The corresponding solutions lose generally their sparsity as the ones derived from optimizing a QP problem. Consequently, these algorithms have poor scalability to the size of the dataset to great extent.

From the horizontal directions in the framework, from EKM to SLMM, the descriptions of the data cluster distribution get finer and finer. As a result, the performance of the classifiers, in general, also improves gradually. The LapSVM characterizes the data manifold distribution. Though the point granularity is finer than the cluster granularity, the classification performance of LapSVM does not always exceed the cluster granularity models. Its performance depends on whether the data structures are more coincidental with the manifold assumption or the cluster assumption, which is a validation for the "No Free Lunch" Theorem [21]. In fact, in the real-world problems, the two learning algorithms are basically comparable, which we will discuss in detail in the experimental section. Furthermore, LapSVM constructs a Laplacian graph for each class, which results in the same number of the manifold regularization terms as the number of classes [7], [22]. Consequently, dependence on the number of given classes makes LapSVM difficult to scale well [23]. The algorithm sometimes involves more computational cost than SLMM in multiclass recognitions, due to the complex adjustment of the free regularization parameters corresponding to the manifold regularization terms.

In summary, the framework not only explores the relationships among these structural SVM-based algorithms but also reveals their characteristics deeply. Although widely applied,

they either characterize the data structure insufficiently or have high computational complexity. This is our major motivation to develop the new large margin classifier model SRSVM. SRSVM follows the same cluster granularity as SLMM, which has shown better performance than SVM, M^4, and radial basis function network [5], but the corresponding optimization problem can still be solved by QP to reduce the computational difficulty and complexity in SLMM. Furthermore, compared to LapSVM, SRSVM only has one cluster regularization term in the objective function, which also lightens the optimization burden in LapSVM greatly.

III. SRSVM

Following the line of the research in the cluster granularity model, SRSVM algorithm has two steps: clustering and learning. SRSVM adopts some clustering techniques to capture the data distribution within classes, and then directly embeds the minimization of the compactness between the estimated clusters into the objective function. Moreover, the algorithm can also be extended to the nonlinear version by the kernel trick. In the following subsections, we will discuss these steps concretely.

A. Clustering

Many clustering methods, such as K-means [24], nearest neighbor clustering [25], and fuzzy clustering [26], can be applied in this step. As in SLMM with the same cluster granularity as our model, clustering is employed to investigate the underlying data distribution within classes in SRSVM. After clustering, the structural information is introduced into the optimization problem 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 [27] which is one of the hierarchical clustering techniques [28].

Concretely, if A and B are two clusters, their Ward's linkage $W(A, B)$ can be calculated as [5]

$$W(A, B) = \frac{|A| \cdot |B| \cdot \|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|^2}{(|A| + |B|)}$$

where $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are the means of the two clusters, respectively.

Initially, each sample is a cluster in the clustering algorithm. The Ward's linkage of two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as $W(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2$ [5]. When two clusters A and B are being merged to a new cluster A', the linkage $W(A', C)$ of A' and other cluster C can be conveniently derived from $W(A, C)$, $W(B, C)$, and $W(A, B)$ by [5]

$$W(A', C) = \frac{(|A| + |C|)\, W(A, C) + (|B| + |C|)\, W(B, C) - |C|\, W(A, B)}{|A| + |B| + |C|}.$$

During clustering, the Ward's linkage between clusters to be merged increases as the number of clusters decreases [5]. We can draw a merge distance curve to represent this process. Here we take one class in Sonar in the UCI database (the UCI Machine Learning Repository) as an example to illustrate the process, as shown in Fig. 4. Salvador and Chan [29] provided a method to automatically determine the number of clusters that selects the number corresponding to the knee point, i.e., the point of maximum curvature, on the curve. Furthermore, the clustering method can also be applicable in the kernel space. For more details, interested readers can refer to the the literature [5].

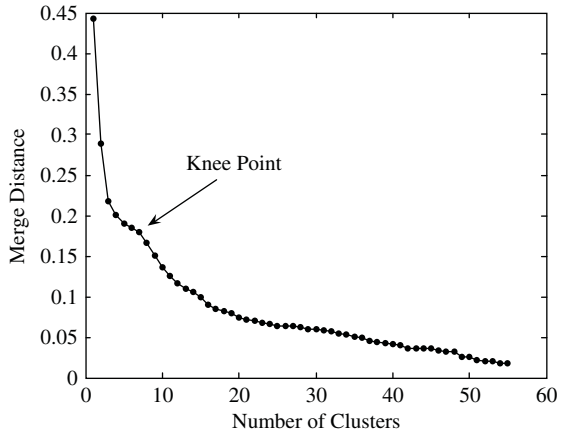

Fig. 4. Choosing the knee point corresponding to the optimal number of clusters in Sonar.

B. Learning

After clustering, we obtain two sets of c_1 and c_2 clusters, respectively, in the two classes. We denote the clusters in the classes as $P_1, \ldots, P_{c_1}$ and $N_1, \ldots, N_{c_2}$ respectively. In Proposition 1, we have validated that SVM gives a natural lower bound to the separability between classes by the constraints. Here we pay more attention to the compactness within classes, i.e., the clusters that cover the different structural information in different classes. We aim to maximize the margin and simultaneously minimize the compactness. Accordingly, the SRSVM model can be formulated as

$$\min_{\boldsymbol{w},b} \frac{\|\boldsymbol{w}\|^2}{2} + \frac{\lambda}{2}\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$$
$$\text{s.t. } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, n \tag{12}$$

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

When the data are linearly nonseparable, SRSVM can further introduce the slack variables ξ_i. The objective function is reformulated as

$$\min_{\boldsymbol{w},b} \frac{\|\boldsymbol{w}\|^2}{2} + \frac{\lambda}{2}\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} + C\sum_{i=1}^{n} \xi_i$$
$$\text{s.t. } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n. \tag{13}$$

Incorporating the constraints into the objective function, we can rewrite (13) as a primal Lagrangian. Then, we transform the primal into the dual problem following the same steps as

SVM:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left[\boldsymbol{x}_i^T (\boldsymbol{I} + \lambda \boldsymbol{\Sigma})^{-1} \boldsymbol{x}_j \right]$$
$$\text{s.t. } 0 \le \alpha_i \le C, \quad i = 1, \ldots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{14}$$

Problem (14) is a typical convex optimization problem. By using the QP techniques, we can obtain the solution α_i. Then, the derived classifier function can be formulated as follows, which is used to predict the class labels for future unseen data $\boldsymbol{x}$:

$$f(\boldsymbol{x}) = sgn \left[\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i^T (\boldsymbol{I} + \lambda \boldsymbol{\Sigma})^{-1} \boldsymbol{x} + b \right]. \tag{15}$$

C. Kernelization

Early in the 1960s, Minsky and Papert [30] had highlighted the limited computational power of linear learning machines [1]. While samples are linearly nonseparable in the input space, the performance of linear classifiers will descend heavily. Kernelization offers an alternative solution by projecting the samples into a high-dimensional kernel space to increase the computational power of the linear classifiers [1]. According to Cover's pattern separability theory, linearly nonseparable samples in the input space may be mapped into a kernel space to make them more likely linearly separable, as long as the mapping is nonlinear and the dimensionality of the kernel space is high enough (even infinity) [5], [31]. However, if the samples are noisy, in general linear separability cannot be guaranteed in the kernel space unless very powerful kernels are used that may lead to overfitting. Two most often used tricks of avoiding overfitting are adoption of the soft margin optimization and regularization. In this subsection, the nonlinear soft margin SRSVM using the kernel trick is developed to further improve the classification performance for complex pattern recognition problems. Furthermore, different from the kernelization in SLMM, SRSVM only needs to kernelize a total covariance matrix obtained by summing all the clusters instead each cluster covariance matrix and, as a result, SRSVM can be implemented more simply and effectively.

Now, assume that a nonlinear (implicit) mapping is $\boldsymbol{\Phi} : \boldsymbol{R}^m \to \boldsymbol{H}$, where $\boldsymbol{H}$ is a Hilbert space with higher dimension. Then the optimization (objective) function of soft margin SRSVM in the kernel space can be described as

$$\min_{\boldsymbol{w},b} \frac{\|\boldsymbol{w}\|^2}{2} + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{\Sigma}^{\boldsymbol{\Phi}} \boldsymbol{w} + C \sum_{i=1}^{n} \xi_i$$
$$\text{s.t. } y_i(\boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}_i) + b) \ge 1 - \xi_i, \xi_i \ge 0, \quad i = 1, \ldots, n \tag{16}$$

where $\boldsymbol{\Sigma}^{\boldsymbol{\Phi}} = \boldsymbol{\Sigma}_{P_1}^{\boldsymbol{\Phi}} + \cdots + \boldsymbol{\Sigma}_{P_{c_1}}^{\boldsymbol{\Phi}} + \boldsymbol{\Sigma}_{N_1}^{\boldsymbol{\Phi}} + \cdots + \boldsymbol{\Sigma}_{N_{c_2}}^{\boldsymbol{\Phi}}$, $\boldsymbol{\Sigma}_{P_i}^{\boldsymbol{\Phi}}$ and $\boldsymbol{\Sigma}_{N_j}^{\boldsymbol{\Phi}}$ denotes the corresponding covariance matrices of the clusters obtained by the kernel Ward's linkage clustering [5], [32] in the kernel space, $i = 1, \ldots, c_1$, $j = 1, \ldots, c_2$.

Then the dual problem is

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j$$
$$\times \left[\boldsymbol{\Phi}(\boldsymbol{x}_i)^T (\boldsymbol{I} + \lambda \boldsymbol{\Sigma}^{\boldsymbol{\Phi}})^{-1} \boldsymbol{\Phi}(\boldsymbol{x}_j) \right]$$
$$\text{s.t. } 0 \le \alpha_i \le C, \quad i = 1, \cdots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{17}$$

However, due to the higher or even infinite dimensions, $\boldsymbol{\Phi}$ cannot often be explicitly formulated. A remedy to this problem is to express all computations in terms of dot products, called the kernel trick [9]. The kernel function $k : \boldsymbol{R}^m \times \boldsymbol{R}^m \to \boldsymbol{R}$, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\Phi}(\boldsymbol{x}_i)^T \boldsymbol{\Phi}(\boldsymbol{x}_j)$ derives the corresponding kernel matrix $\boldsymbol{K} \in \boldsymbol{R}^{n \times n}$,$\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which is called Gram matrix. For more details of kernel tricks, interested readers can refer to the literature [33], [34].

Consequently, we transform (17) into the form of dot products so as to adopt the kernel trick. For each covariance matrix in the kernel space, we have

$$\boldsymbol{\Sigma}_i^{\boldsymbol{\Phi}} = \frac{1}{\left|C_i^{\boldsymbol{\Phi}}\right|} \sum_{\boldsymbol{\Phi}(\boldsymbol{x}_j) \in C_i^{\boldsymbol{\Phi}}} \left[\boldsymbol{\Phi}(\boldsymbol{x}_j) - \boldsymbol{\mu}_i^{\boldsymbol{\Phi}} \right] \left[\boldsymbol{\Phi}(\boldsymbol{x}_j) - \boldsymbol{\mu}_i^{\boldsymbol{\Phi}} \right]^T$$
$$= \frac{1}{\left|C_i^{\boldsymbol{\Phi}}\right|} \boldsymbol{T}_i^{\boldsymbol{\Phi}} \boldsymbol{T}_i^{\boldsymbol{\Phi}^T} - \boldsymbol{T}_i^{\boldsymbol{\Phi}} \vec{\boldsymbol{I}}_{\left|C_i^{\boldsymbol{\Phi}}\right|} \vec{\boldsymbol{I}}_{\left|C_i^{\boldsymbol{\Phi}}\right|}^T \boldsymbol{T}_i^{\boldsymbol{\Phi}^T} \tag{18}$$

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

Then we obtain

$$\boldsymbol{\Sigma}^{\boldsymbol{\Phi}} = \sum_{i=1}^{c_1+c_2} \boldsymbol{\Sigma}_i^{\boldsymbol{\Phi}}$$
$$= \sum_{i=1}^{c_1+c_2} \boldsymbol{T}_i^{\boldsymbol{\Phi}} \boldsymbol{T}_i^{\boldsymbol{\Phi}^T} / \left|C_i^{\boldsymbol{\Phi}}\right| - \boldsymbol{T}_i^{\boldsymbol{\Phi}} \vec{\boldsymbol{I}}_{\left|C_i^{\boldsymbol{\Phi}}\right|} \vec{\boldsymbol{I}}_{\left|C_i^{\boldsymbol{\Phi}}\right|}^T \boldsymbol{T}_i^{\boldsymbol{\Phi}^T} \tag{19}$$
$$= \left[\boldsymbol{T}_1^{\boldsymbol{\Phi}} \cdots \boldsymbol{T}_{c_1+c_2}^{\boldsymbol{\Phi}} \right]$$
$$\times \begin{bmatrix} \boldsymbol{I}_{\left|C_1^{\boldsymbol{\Phi}}\right|} / \left|C_1^{\boldsymbol{\Phi}}\right| - \vec{\boldsymbol{I}}_{\left|C_1^{\boldsymbol{\Phi}}\right|} \vec{\boldsymbol{I}}_{\left|C_1^{\boldsymbol{\Phi}}\right|}^T & & \\ & \ddots & \\ & & \boldsymbol{I}_{\left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right|} / \left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right| - \vec{\boldsymbol{I}}_{\left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right|} \vec{\boldsymbol{I}}_{\left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right|}^T \end{bmatrix}$$
$$\times \begin{bmatrix} \boldsymbol{T}_1^{\boldsymbol{\Phi}^T} \\ \vdots \\ \boldsymbol{T}_{c_1+c_2}^{\boldsymbol{\Phi}^T} \end{bmatrix}$$

where $\boldsymbol{I}_{\left|C_i^{\boldsymbol{\Phi}}\right|}$ is a $\left|C_i^{\boldsymbol{\Phi}}\right| \times \left|C_i^{\boldsymbol{\Phi}}\right|$ identity matrix, $i \in [1, c_1 + c_2]$.

Let the matrices

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{I}_{\left|C_1^{\boldsymbol{\Phi}}\right|} / \left|C_1^{\boldsymbol{\Phi}}\right| - \vec{\boldsymbol{I}}_{\left|C_1^{\boldsymbol{\Phi}}\right|} \vec{\boldsymbol{I}}_{\left|C_1^{\boldsymbol{\Phi}}\right|}^T & & \\ & \ddots & \\ & & \boldsymbol{I}_{\left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right|} / \left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right| - \vec{\boldsymbol{I}}_{\left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right|} \vec{\boldsymbol{I}}_{\left|C_{c_1+c_2}^{\boldsymbol{\Phi}}\right|}^T \end{bmatrix}$$

and $\boldsymbol{P}^{\boldsymbol{\Phi}} = \left[\boldsymbol{T}_1^{\boldsymbol{\Phi}} \quad \cdots \quad \boldsymbol{T}_{c_1+c_2}^{\boldsymbol{\Phi}} \right]$.

Then $\boldsymbol{\Sigma}^{\boldsymbol{\Phi}} = \boldsymbol{P}^{\boldsymbol{\Phi}}\boldsymbol{\Psi}\boldsymbol{P}^{\boldsymbol{\Phi}T}$.

By the Woodbury's formula [35]

$$(\boldsymbol{A}+\boldsymbol{UBV})^{-1} = \boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{UB}(\boldsymbol{B}+\boldsymbol{BVA}^{-1}\boldsymbol{UB})^{-1}\boldsymbol{BVA}^{-1}. \quad (20)$$

So

$$\left(\boldsymbol{I}+\lambda\boldsymbol{\Sigma}^{\boldsymbol{\Phi}}\right)^{-1} = \left(\boldsymbol{I}+\lambda\boldsymbol{P}^{\boldsymbol{\Phi}}\boldsymbol{\Psi}\boldsymbol{P}^{\boldsymbol{\Phi}T}\right)^{-1} = \boldsymbol{I}-\lambda\boldsymbol{P}^{\boldsymbol{\Phi}}\boldsymbol{\Psi}\left(\boldsymbol{\Psi}+\lambda\boldsymbol{\Psi}\boldsymbol{P}^{\boldsymbol{\Phi}T}\boldsymbol{P}^{\boldsymbol{\Phi}}\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}\boldsymbol{P}^{\boldsymbol{\Phi}T}. \quad (21)$$

By substituting (21) into the optimization function (17), we have the kernel form of the dual problem as follows:

$$\begin{aligned}\max_{\boldsymbol{\alpha}} &\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \\ &\times\left[\boldsymbol{K}_{ij}-\lambda\tilde{\boldsymbol{K}}_i^T\boldsymbol{\Psi}\left(\boldsymbol{\Psi}+\lambda\boldsymbol{\Psi}\hat{\boldsymbol{K}}\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}\tilde{\boldsymbol{K}}_j\right] \\ &\text{s.t. } 0\le\alpha_i\le C,\ldots,n \\ &\sum_{i=1}^{n}\alpha_i y_i=0 \end{aligned} \quad (22)$$

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

D. Relationship with SVM and SLMM

In this subsection, we discuss the relationship among SRSVM, SVM, and SLMM, and present how the three algorithms can be transformed to each other under some special conditions. With these analyses, the characteristics of SRSVM can be demonstrated in depth. For simplicity but without loss of generality, we only analyze the linearly separable version.

1) Relationship with SVM: Clearly, if we assume that each ellipsoid cluster is a unit ball, i.e., $\boldsymbol{\Sigma}_{P_i} = \boldsymbol{\Sigma}_{N_j} = \boldsymbol{I}$, $i = 1,\cdots,c_1$, $j = 1,\cdots,c_2$, and $\boldsymbol{I}$ is the identity matrix, the optimization problem of SRSVM can be rewritten as

$$\begin{aligned}&\min_{\boldsymbol{w},b}\frac{\|\boldsymbol{w}\|^2}{2}+\frac{\lambda(c_1+c_2)\|\boldsymbol{w}\|^2}{2} \\ &\text{s.t. } y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\ge 1,\quad i=1,\ldots,n.\end{aligned} \quad (23)$$

Given the regularization parameter λ and the cluster numbers c_1 and c_2, especially taking $\lambda = 0$, (23) can be exactly formulated as the optimization problem (2) in SVM

$$\begin{aligned}&\min_{\boldsymbol{w},b}\frac{\|\boldsymbol{w}\|^2}{2} \\ &\text{s.t. } y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\ge 1,\quad i=1,\ldots,n.\end{aligned}$$

Moreover, the optimization problem (1) of SLMM in linearly separable version is

$$\max\rho \quad (24)$$

$$\text{s.t. } (\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\frac{|P_i|}{Max_P}\rho\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}_{P_i}\boldsymbol{w}},\quad \boldsymbol{x}_l\in P_i, \quad (24.1)$$

$$-(\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\frac{|N_j|}{Max_N}\rho\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}_{N_j}\boldsymbol{w}},\quad \boldsymbol{x}_l\in N_j, \quad (24.2)$$

$$\boldsymbol{w}^T\boldsymbol{r}=1. \quad (24.3)$$

Yeung *et al.* [5] have shown that, if one further assumes that each cluster just contains a single sample, i.e., $C_P = |P|$ and $C_N = |N|$, (24) can be transformed to

$$\begin{aligned}&\max\rho \\ &\text{s.t. } (\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\rho\|\boldsymbol{w}\|, \quad \boldsymbol{x}_l\in P \\ &-(\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\rho\|\boldsymbol{w}\|, \quad \boldsymbol{x}_l\in N \\ &\boldsymbol{w}^T\boldsymbol{r}=1.\end{aligned} \quad (25)$$

Imposing $\rho\|\boldsymbol{w}\| = 1$, instead of the constraint (24.3), (25) can also be formulated the same as (2) in SVM.

In summary, SVM can be viewed as a special case of both SRSVM and SLMM by assuming that each cluster is represented by a unit ball, even a single sample. Therefore, basically, SVM does not embed the data structural information into its own design, which leads to its relatively poor performance than SRSVM and SLMM in the complex classification problems.

2) Relationship with SLMM: SRSVM and SLMM are both cluster granularity models that incorporate the data distribution information in a local way and assume the covariance matrices of the clusters in each class containing the trend of data occurrence in statistics [4]. However, the two algorithms have many various properties also that are embodied not only in the optimization formulation but also in the different emphasis on the utilization of the data structure.

For simplicity, here we assume that the weights for the clusters in SLMM are equal, i.e., $|P_i|/Max_P = |N_j|/Max_N$, $i = 1,\cdots,C_P$, $j = 1,\cdots,C_N$.

Proposition 2: The optimization problem of SLMM can be approximately transformed to the minimization of the covariance matrix sum in (12) of SRSVM.

Proof: Focus on the constraints (24.1) and (24.2) in (24) of SLMM

$$\begin{cases}(\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\rho\sqrt{\boldsymbol{w}^T\Sigma_{P_i}\boldsymbol{w}}, & \boldsymbol{x}_l\in P_i \\ -(\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\rho\sqrt{\boldsymbol{w}^T\Sigma_{N_j}\boldsymbol{w}}, & \boldsymbol{x}_l\in N_j.\end{cases} \quad (26)$$

Relax the functional margin to 1

$$\begin{cases}(\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\rho\sqrt{\boldsymbol{w}^T\Sigma_{P_i}\boldsymbol{w}}\ge 1, & \boldsymbol{x}_l\in P_i \\ -(\boldsymbol{w}^T\boldsymbol{x}_l+b)\ge\rho\sqrt{\boldsymbol{w}^T\Sigma_{N_j}\boldsymbol{w}}\ge 1, & \boldsymbol{x}_l\in N_j.\end{cases} \quad (27)$$

So we have

$$\begin{cases}\frac{1}{\rho}\le\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}_{P_i}\boldsymbol{w}}, & i=1,\cdots,C_P \\ \frac{1}{\rho}\le\sqrt{\boldsymbol{w}^T\boldsymbol{\Sigma}_{N_j}\boldsymbol{w}}, & j=1,\cdots,C_N.\end{cases} \quad (28)$$

Squaring the inequality (28), we further obtain

$$\frac{1}{\rho^2}\le\min\left(\boldsymbol{w}^T\boldsymbol{\Sigma}_{P_i}\boldsymbol{w},\boldsymbol{w}^T\boldsymbol{\Sigma}_{N_j}\boldsymbol{w}\right),\quad i=1,\cdots,C_P,\ j=1,\cdots,C_N. \quad (29)$$

Hence, the maximization of ρ in (24) of SLMM will approximate to

$$\min\left(\min\left(\boldsymbol{w}^T\boldsymbol{\Sigma}_{P_i}\boldsymbol{w}, \boldsymbol{w}^T\boldsymbol{\Sigma}_{N_j}\boldsymbol{w}\right)\right), \quad i=1,\cdots,C_P,\ j=1,\cdots,C_N. \tag{30}$$

For $\boldsymbol{\Sigma}_{P_i}$ and $\boldsymbol{\Sigma}_{N_j}$ are symmetric positive semidefinite, we have

$$\begin{aligned}&\min\left(\boldsymbol{w}^T\boldsymbol{\Sigma}_{P_i}\boldsymbol{w}, \boldsymbol{w}^T\boldsymbol{\Sigma}_{N_j}\boldsymbol{w}\right)\\ &\le \frac{\left[\boldsymbol{w}^T\left(\boldsymbol{\Sigma}_{P_1}+\cdots+\boldsymbol{\Sigma}_{P_{C_P}}+\boldsymbol{\Sigma}_{N_1}+\cdots+\boldsymbol{\Sigma}_{N_{C_N}}\right)\boldsymbol{w}\right]}{(C_P+C_N)}\\ &= \frac{\boldsymbol{w}^T\boldsymbol{\Sigma}w}{(C_P+C_N)}\end{aligned} \tag{31}$$

where $\boldsymbol{\Sigma}=\boldsymbol{\Sigma}_{P_1}+\cdots+\boldsymbol{\Sigma}_{P_{C_P}}+\boldsymbol{\Sigma}_{N_1}+\cdots+\boldsymbol{\Sigma}_{N_{C_N}}$.

So the optimization problem in SLMM can be approximately reformulated as

$$\min_{\boldsymbol{w},b}\ \boldsymbol{w}^T\boldsymbol{\Sigma}w \quad \text{s.t. } \boldsymbol{w}^T\boldsymbol{r}=1 \tag{32}$$

which is similar to the minimization of the covariance matrix sum in the optimization problem (12) of SRSVM. ■

Different from SVM, which focuses on the separability between the classes, SLMM emphasizes more the compactness between the clusters within the individual classes. In other words, SLMM maximizes the margin by restricting the compactness. However, this is likely to be insufficient for classifier design. Consequently, to a great extent, SRSVM not only embeds the compactness within the individual classes into the objective function but also integrates the constraints in the SVM formulation to introduce the natural lower bound of the separability between the classes, which may lead to better classification and generalization performance than SLMM. We will address these issues in more detail in the experimental section.

IV. Experiments

To evaluate the proposed SRSVM algorithm, in this section we perform a series of experiments systematically on both toy and real-world classification problems. First, we present a synthetic XOR dataset for clearly comparing SRSVM with SLMM and SVM. On real-world problems, several datasets in the UCI database (the UCI Machine Learning Repository) are used to evaluate the classification accuracies derived from SRSVM in comparison to the other algorithms in the proposed structural large margin classifier framework. Finally, we further apply SRSVM for the image recognition problems.

Due to the better performance of the kernel version, throughout the experiments we uniformly compare the algorithms in the kernel and soft margin cases. The width parameter in the Gaussian kernel and the regularization parameters such as C and λ in the algorithms are selected from the set $\{2^{-10}, 2^{-9}, \cdots, 2^9, 2^{10}\}$ by cross validation. For multiclass cases, we adopt the one-against-all strategy [21] for all the classifiers. We apply sequential minimal optimization (SMO) algorithm [1] to solve the QP problem in SVM, EKM, LapSVM, and SRSVM. Meanwhile, the Sedumi toolbox [36] is used to solve the SOCP optimization in SLMM and the SDP optimization in EKM. All the experiments are performed on a server with Xeon(R) X5460 3.16-GHz processor and 32 766-MB RAM.

TABLE I
Attributes of the Toy XOR Dataset. Each Class Contains Two Clusters

		Probability	Mean	Covariance
Class I	Gaussian distribution I_1	1/2	[2, 5]	[0.75, 0; 0, 5]
	Gaussian distribution I_2	1/2	[1, -5]	[6, 0; 0, 0.75]
Class II	Gaussian distribution II_1	1/2	[-5, 0]	[0.75, 0; 0, 6]
	Gaussian distribution II_2	1/2	[8, 0]	[5, 0; 0, 0.75]

TABLE II
Training and Testing Accuracies (%) of SVM, SLMM, and SRSVM on the Toy XOR Dataset in Cases from 10% to 50% of the Samples in Each Distribution as the Training Sets

Percents of samples	Training accuracy /Testing accuracy		
	SVM	SLMM	SRSVM
10	100.00/92.78	100.00/96.39	100.00/***99.44***
20	99.38/96.09	99.38/97.50	99.38/***99.53***
30	99.58/99.11	99.58/99.46	99.58/***99.64***
40	99.69/99.17	99.69/99.38	99.69/***99.58***
50	99.50/99.50	99.50/***99.75***	99.50/***99.75***

A. Toy Dataset

The XOR problem is a typical linearly nonseparable problem in classification. The toy 2-D dataset is randomly generated under two Gaussian distributions in each class. Table I describes the corresponding attributes of the dataset. Each Gaussian distribution has 200 samples. And the samples in each class are designed to scatter in two clusters I_1, I_2 and II_1, II_2. In order to conduct the comparisons more efficiently, we randomly select 10%, 20%, 30%, 40%, and 50% of the samples in each distribution as the training sets, and the remaining ones as the testing sets.

We compare SRSVM with SLMM and SVM. Due to limited space, we show the resulting discriminant boundaries in cases of 10% and 50% both in training and testing sets as representatives in Figs. 5 and 6, where '*' and '·' denote the samples in the two classes, respectively. We can see that class I has the vertical distribution and class II has a horizontal one. In these cases, the structural information within the classes may be more important than the discriminative information between the classes. The corresponding accuracies in all cases from 10% to 50% are listed in Table II. From the results, we can infer the following.

1) Due to the formal neglect of the structural information within the classes, SVM cannot differentiate the different data occurrence trends, i.e., the two clusters here in each

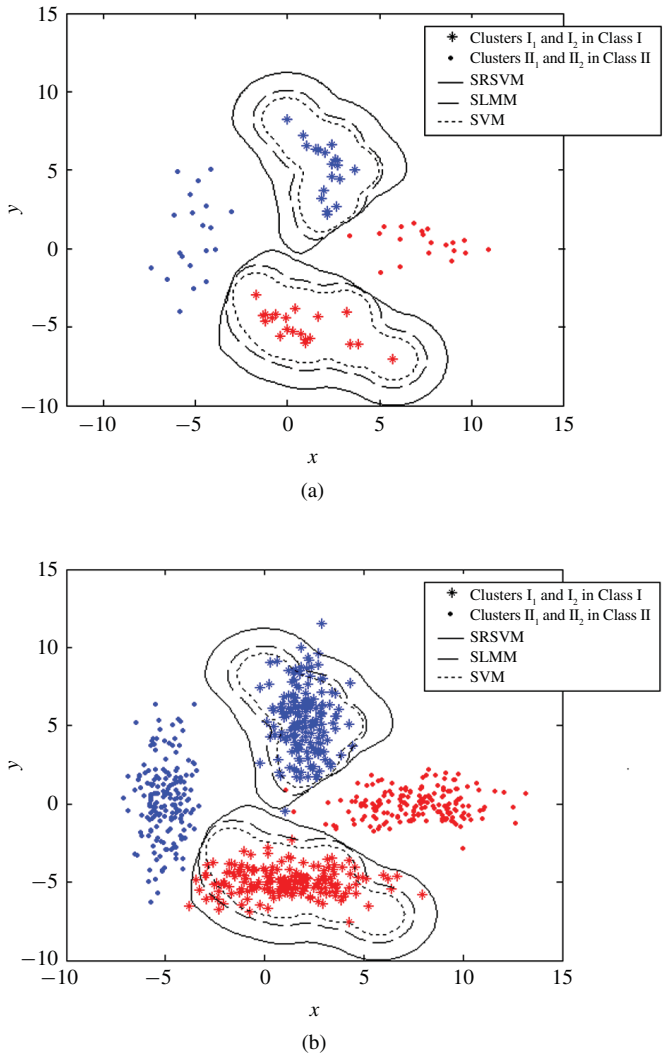

Fig. 5. Classification results of SRSVM, SLMM, and SVM on the toy XOR dataset with 10% of the samples in each distribution as the training set. (a) Discriminant boundaries in the training set. (b) Discriminant boundaries in the testing set.

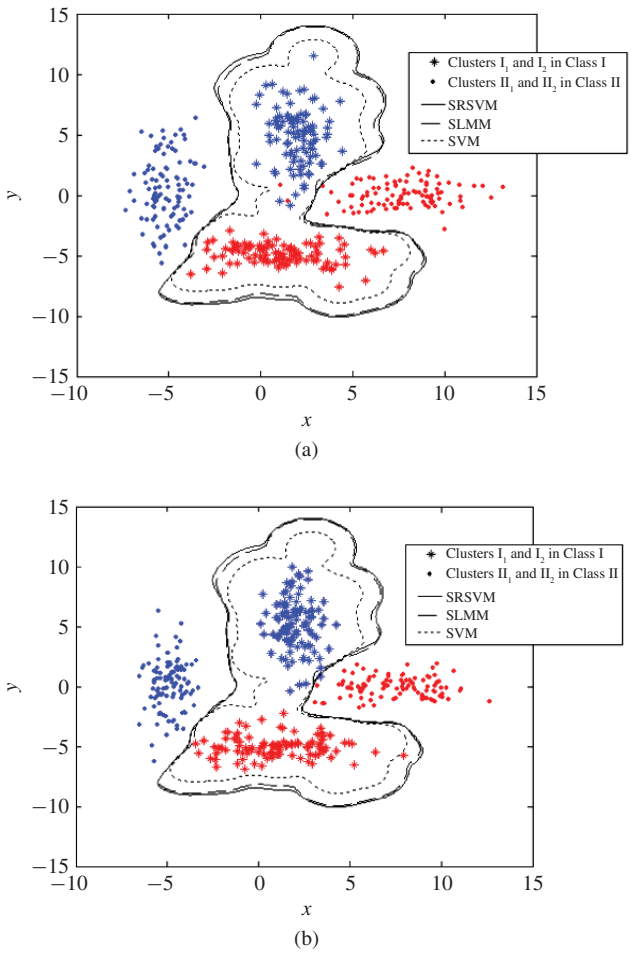

Fig. 6. Classification results of SRSVM, SLMM, and SVM on the toy XOR dataset with 50% of the samples in each distribution as the training set. (a) Discriminant boundaries in the training set. (b) Discriminant boundaries in the testing set.

class. The derived boundaries always approximately lie in the middle of the support vectors [4]–[6] in the training sets in the two cases, which only focus on the separability between the classes. Consequently, though SVM can achieve comparable training accuracies in the training sets and its testing accuracies can increase successively in cases from 10% to 50% with the growth of the training data, it still has relatively poor performance in the testing sets.

2) SLMM exploits the structural information within the classes uncovered by some clustering algorithm, thus it has better classification performance than SVM. From Figs. 5 and 6, its discriminant boundaries basically enclose those of SVM, meaning that SLMM has better generalization performance than SVM. However, owing to the lack of sufficient emphasis on the separability between the classes, SLMM also gives a worse performance than SRSVM in almost all the testing sets except in case of the 50% set which is due to sufficient sampling of the training data.

3) Thanks to capturing the cluster structures in the classes and considering the separability between the classes as well as the compactness within the classes simultaneously, SRSVM gets more reasonable discriminant boundaries than both SLMM and SVM which basically accord with the data occurrence trend, and thus has the best classification performance in all the testing sets. Furthermore, from Figs. 5 and 6, the boundaries of SRSVM enclose those of not only SVM but also SLMM. Especially in case of the 50% case, though SRSVM and SLMM have the same training and testing accuracies, the boundary of SRSVM still almost encloses that of SLMM. As a result, SRSVM can classify more testing data correctly than SLMM and SVM, which further validates that SRSVM actually has better generalization ability.

B. UCI Dataset

To further investigate the effectiveness of our SRSVM, we also evaluate its performance on several real-world datasets in the UCI database, whose attributes are presented in Table III.

TABLE III
ATTRIBUTES OF THE 18 DATASETS IN THE UCI DATABASE

Dataset	Feature	Data	Class
Automobile	25	159	2
Bupa	6	345	2
Hepatitis	19	155	2
Ionosphere	34	351	2
Pima	8	768	2
Sonar	60	208	2
Water	38	86	2
Wdbc	30	569	2
Iris	4	150	3
Tae	5	151	3
New_thyriod	5	215	3
Cmc	9	1473	3
Balance_scale	4	625	3
Vehicle	18	846	4
Dermatology	33	366	6
Ecoli	6	332	6
Glass	9	214	6
Yeast	8	1484	10

For the multiclass datasets, we adopt the one-against-all strategy [21] uniformly in all the classifiers. Since SLMM have been shown to be better than M^4 in terms of classification accuracy [5], in this experiment we systematically compare SRSVM with the other algorithms in the proposed framework, i.e., SVM, EKM, SLMM, and LapSVM.

For each dataset, we divide randomly the samples into two nonoverlapping training and testing sets, where each set contains almost half of samples in each class. This process is repeated 10 times to generate 10 independent runs for each dataset; their average results are reported in Table IV.

We also compare the average running times and the average support vector numbers of all the algorithms after the cross validations of the parameters, as shown in Tables V and VI. The intermediate numbers of clusters obtained by the Ward's linkage clustering algorithm used in SLMM and SRSVM are listed in Table VII, where the number in the first row denotes the total cluster number in the two class and the numbers in the bracket in the second row denote the cluster numbers in the respective classes. For multiclass datasets, we use the one-against-all strategy [21] to establish multiclass classifiers. Such an establishment generally leads to inexact estimation for the numbers of support vectors and clusters in each individual class to some extent and, consequently, we just report the comparison results conducted on the two-class datasets.

From these results, we can make several interesting observations as follows.

1) EKM, SLMM, LapSVM, and SRSVM basically have the better classification performance than SVM on the overall datasets. As the improved algorithms of SVM, they devote to embedding the data structure into the traditional SVM framework. According to the well-known "No Free Lunch" Theorem [21], the introduction of as much prior knowledge into data as possible can indeed improve the classifier performance. As a result, the outstanding performance of these algorithms further validates the necessity of structural information as prior knowledge for the classifier design.
2) For the three cluster-based algorithms, EKM emphasizes on the data global granularity and neglects the local structures. Consequently, its capability is weaker than the other two algorithms with finer granularity. SLMM and SRSVM both adopt the Ward's linkage clustering algorithm to capture the corresponding cluster structures in individual classes. As shown in Table VII, the clustering algorithm obtains relatively reasonable clusters both in the small and large datasets. As a result, by embedding such structural information, SLMM and SRSVM achieve better performance than EKM. Furthermore, on almost all the datasets except Pima, SRSVM outperforms SLMM due to its more consideration of data distribution within and between classes. The gap of their classification accuracies on Pima is less than 1%.
3) LapSVM is a manifold-based algorithm that focuses on the point granularity and introduces the data manifold structure into the classifier design. In the experiments, its performance is comparable to SRSVM, which further demonstrates that the cluster and manifold assumptions about data structure are both reasonable in the real-world problems. When the data structure is closer to the manifold geometry, LapSVM is relatively dominant. Otherwise, SRSVM is ascendant. Therefore, how to select the suitable classifier in the applications actually depends on more prior knowledge about the data.
4) For comparing efficiency and solution sparsity of these algorithms, Tables V and VI give their average running times and the average support vector numbers, respectively, on all the two-class datasets. From Table V, we observe that SLMM has the highest training times due to its implementation of SOCP in solving its dual problem. EKM is the second highest since it first adopts SDP in estimating the minimum volume bounding ellipsoid surrounding the data and then still uses QP to solve a transformative SVM problem after remapping the data to a unit sphere by the estimated centroid and covariance matrix of the ellipsoid. SVM, LapSVM, and SRSVM have far lower training times than SLMM and EKM since their optimization problems can be directly solved by QP. Among the three algorithms, SRSVM has the highest training times due to the clustering process, and LapSVM is the second highest for the constructions of the neighbor graph on the data and the corresponding Laplacian matrix. Furthermore, as the improved methods, SRSVM, LapSVM, and EKM also have better sparsity than SVM according to the reported average numbers of support vectors in Table VI, which means that these algorithms more likely have better scalability to the size of the datasets in the real applications. On the contrary, the optimization problem of SLMM can only boil down to SOCP and thus the solutions lose the sparsity.
5) In order to find out whether SRSVM is significantly better than the other algorithms, we perform the t-test on the classification results of the 10 runs to calculate the

TABLE IV
CLASSIFICATION ACCURACIES (%) AND VARIANCES COMPARED BETWEEN SVM, EKM, SLMM, LAPSVM, AND SRSVM ON THE UCI DATASETS

Dataset	Classification Accuracy				
	SVM	EKM	SLMM	LapSVM	SRSVM
Automobile	88.48* ± 2.49	88.53* ± 2.22	88.63* ± 2.73	87.85* ± 3.00	***91.14*** ± 2.46
Bupa	73.06* ± 2.43	73.36* ± 2.35	73.52* ± 2.85	***78.72**** ± 2.65	76.18 ± 2.76
Hepatitis	79.61* ± 0.63	79.68* ± 1.94	79.82* ± 2.70	78.18* ± 2.99	***83.25*** ± 2.63
Ionosphere	95.11* ± 1.34	95.34* ± 0.99	95.63* ± 1.63	***98.30*** ± 1.76	97.52 ± 1.52
Pima	77.08* ± 1.26	77.35 ± 1.52	***79.46*** ± 1.69	77.88 ± 1.89	78.63 ± 1.79
Sonar	85.00* ± 3.55	86.15 ± 3.63	86.21 ± 3.09	***91.26**** ± 2.80	87.60 ± 2.74
Water	90.51* ± 2.90	94.73* ± 1.14	95.49* ± 3.46	98.21 ± 2.18	***98.69*** ± 1.40
Wdbc	94.25* ± 0.53	94.34* ± 0.64	94.57* ± 1.33	94.84* ± 0.88	***95.72*** ± 0.72
Iris	98.27 ± 1.10	98.40 ± 2.18	98.47 ± 1.59	98.67 ± 2.22	***98.83*** ± 1.78
Tae	54.34 ± 3.72	55.53 ± 4.18	55.56 ± 3.57	***57.63*** ± 4.44	56.18 ± 3.63
New_thyriod	96.02 ± 1.38	96.30 ± 2.31	96.51 ± 2.22	***97.30*** ± 2.04	97.23 ± 2.17
Cmc	56.41 ± 1.63	56.25 ± 1.72	56.29 ± 1.54	56.46 ± 1.89	***56.79*** ± 1.33
Balance_scale	92.04* ± 0.46	92.21* ± 0.52	92.63* ± 0.99	92.77* ± 0.64	***93.54*** ± 0.90
Vehicle	74.76* ± 1.37	75.35* ± 1.79	76.76* ± 2.04	75.87* ± 2.36	***78.70*** ± 1.99
Dermatology	97.28* ± 0.54	97.75 ± 1.69	97.77 ± 1.96	***98.78*** ± 1.43	98.51 ± 1.88
Ecoli	89.17* ± 1.34	90.33 ± 1.02	90.43 ± 2.03	89.19 ± 2.65	***90.54*** ± 1.59
Glass	72.75* ± 3.96	73.10 ± 3.68	74.73 ± 3.85	75.57 ± 3.57	***76.04*** ± 4.04
Yeast	60.58 ± 1.50	60.82 ± 1.35	60.96 ± 1.87	***61.70*** ± 1.63	61.21 ± 1.79

'*' Denotes that the difference between SRSVM and the other four algorithms is significant at 5% significance level, i.e., t-value >1.7341

statistical significance of SRSVM. The null hypothesis H_0 demonstrates that there is no significant difference between the mean numbers of patterns correctly classified by SRSVM and the other algorithms. 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 IV will be denoted "*." Consequently, as shown in Table IV, it can be clearly found that SRSVM possesses significantly superior classification performance compared with SVM, EKM, and SLMM on the most datasets. Meanwhile, SRSVM and LapSVM are basically comparable on the datasets. This just accords with our conclusions.

C. Image Recognition

Image classification is a popular problem in pattern recognition [23], [37]–[39]. In this subsection, we also apply our proposed method to image recognition. Two well-known and publicly available databases corresponding to typical image classification problems, i.e., objects (COIL-20) and handwritten digits (USPS), are used to evaluate SRSVM with SVM, EKM, SLMM, and LapSVM.

Fig. 7. Illustration of 20 subjects on the COIL-20 database.

1) Dataset Description and Experimental Setting: COIL-20 [40] is a database of gray-scale images of 20 objects, as shown in Fig.7 [39]. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360° to vary the object poses with respect to a fixed camera. Images of the objects were taken at pose intervals of 5°, which

TABLE V
AVERAGE RUNNING TIMES (S) COMPARED BETWEEN SVM, EKM, SLMM, LAPSVM AND SRSVM ON THE TWO-CLASS UCI DATASETS

Dataset	Training time/Testing time				
	SVM	EKM	SLMM	LapSVM	SRSVM
Automobile	0.47 / 0.02	5.83 / 0.02	28.08 / 0.05	0.86 / 0.03	2.20 / 0.05
Bupa	0.50 / 0.03	85.47 / 0.03	417.05 / 0.03	1.58 / 0.03	5.12 / 0.03
Hepatitis	0.42 / 0.03	4.69 / 0.03	39.33 / 0.05	0.80 / 0.03	2.17 / 0.05
Ionosphere	0.58 / 0.06	122.56 / 0.03	198.03 / 0.03	1.20 / 0.03	4.06 / 0.08
Pima	0.72 / 0.06	480.09 / 0.13	620.86 / 0.15	3.90 / 0.22	10.97 / 0.20
Sonar	0.50 / 0.05	17.47 / 0.05	41.20 / 0.03	0.89 / 0.02	3.03 / 0.03
Water	0.30 / 0.08	3.80 / 0.25	13.09 / 0.08	2.73 / 0.08	2.50 / 0.05
Wdbc	0.47 / 0.03	336.36 / 0.03	454.21 / 0.05	3.62 / 0.25	12.73 / 0.12
Average training time (s.)	0.50	132.03	226.48	1.95	5.35
Average Testing Time (s)	0.05	0.07	0.06	0.08	0.08

TABLE VI
AVERAGE SUPPORT VECTOR NUMBERS COMPARED BETWEEN SVM, EKM, SLMM, LAPSVM, AND SRSVM ON THE TWO-CLASS UCI DATASETS

Dataset	Number of support vectors				
	SVM	EKM	SLMM	LapSVM	SRSVM
Automobile	59	61	—	44	45
Bupa	113	100	—	110	100
Hepatitis	62	62	—	62	62
Ionosphere	123	113	—	90	92
Pima	273	268	—	259	250
Sonar	62	55	—	75	55
Water	32	32	—	32	32
Wdbc	203	178	—	185	178

Fig. 8. Illustration of 10 digits on the USPS database.

corresponds to 72 images per object. For our experiments, we have resized each of the original 1440 images down to 32×32 pixels. We partition the database into the different gallery and probe sets where Gm/Pn indicates that m images per object randomly selected for training and the remaining n images are used for testing [41].

The USPS database consists of grayscale handwritten digit images from 0 to 9, as shown in Fig. 8 [42]. Each digit contains 1100 images, and the size of each image is 16×16 pixels with 256 gray levels. Due to the large scale data, here we randomly choose 10%, 20%, and 30% per digit for training and the remaining for testing.

2) Evaluation of Classification Performance: Fig. 9 shows the experimental results of the five algorithms on the COIL-20 and USPS databases, respectively, in terms of different sampling in the training and testing sets. From these results, we can also obtain several attractive insights as follows.

1) The COIL-20 database is a typical pose estimation dataset, where the object images have underlying invariant and associated transformations, such as shift and rotation. Ghodsi *et al*. [43] have shown that such image data naturally imply a low-dimensional intrinsic manifold on which the neighboring samples are small transformations of one another. Consequently, the classification accuracies of LapSVM are better than those of the other algorithms in the G9/P63 and G18/P54 cases. However, with the increase of the training samples, the difference between the accuracies of the five algorithms becomes much smaller, especially in the G18/P54 case.
2) On the USPS database, from 10% to 30% cases, the accuracies of the five algorithms increase steadily with the growth of the training samples. Especially, SRSVM has obvious superiority to the other algorithms in the whole cases, which more likely implies that the samples of the 10 digits tend to the cluster distribution rather than the manifold distribution in the high-dimensional space.

V. GENERALIZATION BOUND ANALYSIS

In this section, we discuss the generalization bounds for the large margin classifiers. In traditional SVMs, we are accustomed to carrying out generalization bound estimation based on the radius of the smallest enclosing sphere of the data and the observed margin on the training set [11], [44]. However, this approach completely ignores the information about the distribution of the data [11]. Therefore, here we adopt another generalization bound for the classifiers, which

TABLE VII
NUMBERS OF CLUSTERS OBTAINED BY THE WARD'S LINKAGE CLUSTERING ALGORITHM USED IN SLMM AND SRSVM ON THE TWO-CLASS UCI DATASETS

	Auto.	Bupa	Hepa.	Iono.	Pima	Sonar	Water	Wdbc
No. of Cluster	7 (4, 3)	12 (6, 6)	8 (3, 5)	18 (10, 8)	15 (11, 4)	11 (4, 7)	7 (4, 3)	16 (8, 8)

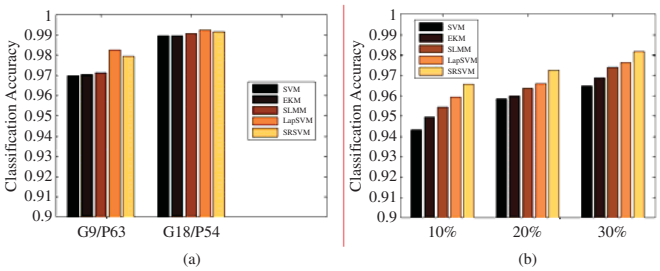

Fig. 9. Classification accuracies compared between SVM, EKM, SLMM, LapSVM, and SRSVM on (a) COIL-20 database and (b) USPS database.

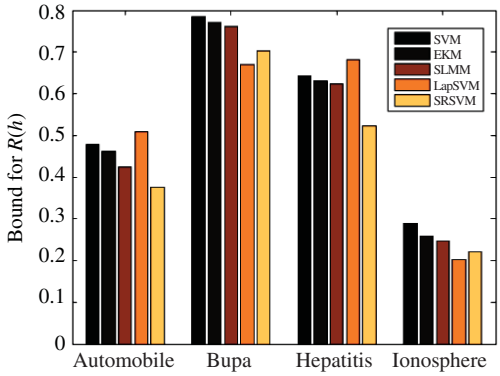

Fig. 10. Bounds of the expected risks for SVM, EKM, SLMM, LapSVM, and SRSVM on the four UCI datasets.

is given by the empirical covering number in terms of the distribution of the eigenvalues of the kernel matrix [11]–[13].

Assume that the data $(\boldsymbol{x}, y) \in \boldsymbol{Z}$ follow a certain distribution $P(\boldsymbol{x}, y)$. The expected risk of a hypothesis $h \in \boldsymbol{F}$ is given by $R(h) = \sum_{(\boldsymbol{x},y)\in\boldsymbol{Z}} \delta(yh(\boldsymbol{x}) \le 0) P(\boldsymbol{x}, y)$ [13]. Given a training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ from $\boldsymbol{Z}$, the empirical margin risk for a certain margin γ is defined by the rate of the samples with $y_i h(\boldsymbol{x}_i) < \gamma$: $R_s^\gamma(h) = (1/n) \sum_{i=1}^{n} \delta(y_i h(\boldsymbol{x}_i) < \gamma)$ [13]. Then the following theorem gives an upper bound for the expected risk [11]–[13].

Theorem 1: Let $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$ be the eigenvalues of the kernel matrix derived from the training samples. Consider the hypothesis class $\boldsymbol{F}(c)_B = \{\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b : \|\boldsymbol{w}\| \le c, |b| \le B\}$. Then the following inequality holds simultaneously for all $\gamma \in (8\Upsilon(m), 1]$:

$$\begin{aligned} &\boldsymbol{P}_{s\in\boldsymbol{Z}^n}\left(\exists h \in \boldsymbol{F}(c)_B : R(h) \ge R_s^r(h)\right. \\ &\left. +\sqrt{(m\ln 2 + ln(\lceil c\rceil/\theta\gamma)\lceil 8B/\gamma\rceil)/(2n)}\right) \\ &\le \theta \end{aligned} \tag{33}$$

where

$$\begin{aligned} \Upsilon(m) &= \min_{j\in\{1,\ldots,m-1\}} 6\times 2^{-\frac{j-1}{k(2^{j-1})}}\left(\lambda_1\cdots\lambda_{k(2^{j-1})}\right)^{\frac{1}{2k(2^{j-1})}} c(m,j) \\ k(l) &= \min\left\{k \in \{1,\cdots,n\} : \lambda_{k+1} \le (\lambda_1\cdots\lambda_k/l^2)^{\frac{1}{k}}\right\} \\ c(m,j) &= \min\left(1, 1.86\sqrt{\log_2(n/(m-j)+1)/(m-j)}\right). \end{aligned} \tag{34}$$

We select four datasets in the UCI database to estimate the generalization bounds for the five classifiers. The corresponding results are shown in Fig. 10. The bounds of SRSVM are smaller than those of SVM, EKM, and SLMM on the four datasets. And the bounds of LapSVM are basically comparable with SRSVM. However, due to the more free regularization parameters involved than the other algorithms and the insufficient samples to characterize the data manifold structure, LapSVM yields unstable capability and its bounds achieve even the largest ones on the Automobile and Hepatitis datasets. These theoretical results are also consistent with the experimental results in the previous section.

VI. CONCLUSION

In this paper, with systematic analysis on the modern large margin classifiers, we first introduced the concept "structural granularity," which characterizes a series of data structures involved in the various classifier design ideas. Based on the different granularities and the formulations for optimization problems, we further constructed a uniform structural framework for these classifiers. A novel algorithm SRSVM was then derived from the cluster granularity in the framework, which captures the data structural information within individual classes by some clustering strategies. Owing to the insights into both SVM and SLMM, we simultaneously embedded the compactness within classes into the objective function as well as the separability between classes into the constraints of SRSVM based on the learning framework of SVM. The corresponding optimization problem follows the same QP formulation as SVM, rather than the SOCP in the related algorithms such as MPM, M^4, and SLMM. As a result, SRSVM not only has much lower computational complexity but also holds the sparsity of the solution. Furthermore, we also discussed the generalization bounds for these algorithms by using the distribution of the eigenvalues of the kernel matrix. The experimental results demonstrated the superiority of our proposed SRSVM compared to the state-of-the-art algorithms in the framework.

There are several directions of future study.

1) Additional generalization: The proposed structural framework still has many parts for further research. The

combinations of different granularities, such as cluster granularity and point granularity, may lead to a large family of new algorithms and we believe that there should be many interesting observations that can be obtained.

2) Large-scale problem: In the experiments, we apply SRSVM in the middle-scale classification problems. However, due to the requirements of the practical applications, large-scale problem solution has become a hot issue in machine learning. Tsang *et al.* [45] have presented an algorithm of ball vector machine (BVM) to improve SVM in the large-scale cases. How to develop a fast algorithm for SRSVM to solve large-scale problems is another interesting topic for future study.

3) Prior knowledge: The experimental results have shown that the cluster granularity algorithm SRSVM and the point granularity algorithm LapSVM have comparable performance in the real-world problems under study here. How to select a suitable classifier between the two algorithms actually depends on more prior knowledge about the data. Consequently, we intend to develop more effective methods to reveal more prior knowledge hidden in the data to guide classifier design.

REFERENCES

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[2] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[3] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[4] K. Huang, H. Yang, I. King, and M. R. Lyu, "Learning large margin classifiers locally and globally," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, 2004, pp. 1–8.

[5] D. Yeung, D. Wang, W. Ng, E. Tsang, and X. Zhao, "Structured large margin machines: Sensitive to data distributions," *Mach. Learn.*, vol. 68, no. 2, pp. 171–200, 2007.

[6] H. Xue, S. Chen, and Q. Yang, "Structural support vector machine," in *Proc. 15th Int. Symp. Neural Netw.*, LNCS 5263. 2008, pp. 501–511.

[7] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from examples," Dept. Comput. Sci., Univ. Chicago, Chicago, IL, Tech. Rep. TR-2004-06, Aug. 2004.

[8] P. Rigollet, "Generalization error bounds in semi-supervised classification under the cluster assumption," *J. Mach. Learn. Res.*, vol. 8, pp. 1369–1392, Jul. 2007.

[9] P. K. Shivaswamy and T. Jebara, "Ellipsoidal kernel machines," in *Proc. 12th Int. Workshop Artif. Intell. Stat.*, 2007, pp. 1–8.

[10] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan, "A robust minimax approach to classfication," *J. Mach. Learn. Res.*, vol. 3, pp. 555–582, Dec. 2002.

[11] B. Schölkopf, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Generalization bounds via eigenvalues of the gram matrix," ESPRIT, Ratingen, Germany, Tech. Rep. NC2-TR-1999-035, Mar. 1999.

[12] R. Kroon, "Support vector machines, generalization bounds and transduction," M.S. thesis, Dept. Comput. Sci., Stellenbosch Univ., Stellenbosch, South Africa, 2003.

[13] H. Kashima, S. Oyama, Y. Yamanishi, and K. Tsuda, "On pairwise kernels: An efficient alternative and generalization analysis," in *Proc. 13th Pacific-Asia Conf. Knowl. Disc. Data Mining*, 2009, pp. 27–30.

[14] M. M. Adankon, M. Cheriet, and A. Biem, "Semisupervised least squares support vector machine," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1858–1870, Dec. 2009.

[15] X. Liang, "An effective method of pruning support vector machine classifiers," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 26–38, Jan. 2010.

[16] B. Liu, Z. Hao, and E. C. C. Tsang, "Nesting one-against-one algorithm based on SVMs for pattern classification," *IEEE Trans. Neural Netw.*, vol. 19, no. 12, pp. 2044–2052, Dec. 2008.

[17] X. Liang, R.-C. Chen, and X. Guo, "Pruning support vector machines without altering performances," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1792–1803, Oct. 2008.

[18] H. Chen, P. Tino, and X. Yao, "Probabilistic classification vector machines," *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 901–914, Jun. 2009.

[19] S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, "A convergent hybrid decomposition algorithm model for SVM training," *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 1055–1060, Jun. 2009.

[20] A. Moore. *Gaussian Mixture Models* [Online]. Available: http://www.autonlab.org/tutorials/gmm.html

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.

[22] M. Belkin, P. Niyogi, and V. Sindhwani, "On mainfold regularization," in *Proc. 10th Int. Workshop Artif. Intell. Stat.*, Bridgetown, Barbados, 2005, pp. 17–24.

[23] H. Xue, S. Chen, and Q. Yang, "Discriminatively regularized least-squares classification," *Pattern Recognit.*, vol. 42, no. 1, pp. 93–104, Jan. 2009.

[24] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[25] S.-Y. Lu and K. S. Fu, "A sentence-to-sentence clustering procedure for pattern analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. 8, no. 5, pp. 381–389, May 1978.

[26] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.

[27] J. H. Ward, "Hierarchical grouping to optimize an objective function," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 236–244, Mar. 1963.

[28] A. K. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[29] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *Proc. 16th IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2004, pp. 576–584.

[30] M. L. Minsky and S. A. Papert, *Perceptrons*. Cambridge, MA: MIT Press, 1969.

[31] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Beijing, China: Tsinghua Univ. Press, 2001.

[32] D. Wang, D. S. Yeung, and E. C. C. Tsang, "Structured one-class classification," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 36, no. 6, pp. 1283–1294, Dec. 2006.

[33] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[34] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1000–1017, Sep. 1999.

[35] M. A. Woodbury, "Inverting modified matrices," Stat. Res. Group, Princeton Univ., Princeton, NJ, Memo. Rep. 42, 1950.

[36] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11, no. 12, pp. 625–653, 1999.

[37] W. W. Y. Ng, A. Dorado, D. S. Yeung, W. Pedrycz, and E. Izquierdo, "Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error," *Pattern Recognit.*, vol. 40, no. 1, pp. 19–32, 2007.

[38] Y. Li, L. O. Shapiro, and J. A. Bilmes, "A generative/discriminative learning algorithm for image classification," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2. Beijing, China, Oct. 2005, pp. 1605–1612.

[39] T. Sun and S. Chen, "Locality preserving CCA with applications to data visualization and pose estimation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 531–543, May 2007.

[40] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Dept. Comput. Sci., Columbia Univ., New York, NY, Tech. Rep. CUCS-005-96, Feb. 1996.

[41] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[42] A. Argyriou, M. Herbster, and M. Pontil, "Combing graph Laplacians for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst. 18*, 2005, pp. 1–8.

[43] A. Ghodsi, J. Huang, F. Southey, and D. Schuurmans, "Tangent-corrected embedding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 518–525.

[44] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *Proc. 1st Int. Conf. Knowl. Discov. & Data Mining*, 1995, pp. 252–257.

[45] I. W. Tsang, A. Kocsor, and J. T. Kwok, "Simpler core vector machines with enclosing balls," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, 2007, pp. 1–8.

Hui Xue received the B.S. degree in mathematics from Nanjing Normal University, Nanjing, China, in 2002, the M.S. degree in mathematics from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, in 2005, and the Ph.D. degree in computer application technology from NUAA in 2008.

She has been with the School of Computer Science and Engineering, Southeast University, Nanjing, as a University Instructor, since 2009. Her current research interests include pattern recognition, machine learning, and neural computing.

Songcan Chen received the B.S. degree in mathematics from Hangzhou University (now merged into Zhejiang University), Zhejiang, China, in 1983, the M.S. degree in computer applications from Shanghai Jiaotong University, Shanghai, China, in 1985, and the Ph.D. degree in communication and information systems from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1997.

He began working at the NUAA in January 1986. Since 1998, he has been a full-time Professor with the Department of Computer Science and Engineering at NUAA. He has authored or co-authored over 130 scientific peer-reviewed papers. His current research interests include pattern recognition, machine learning, and neural computing.

Qiang Yang (F'09) received the Bachelors degree in astrophysics from Peking University, Beijing, China, and the Ph.D. degree in computer science from the University of Maryland, College Park.

He is currently a Faculty Member in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. His current research interests include data mining and machine learning, artificial intelligence planning, and sensor-based activity recognition.

Dr. Yang is a member of the Association for the Advancement of Artificial Intelligence and Association for Computing Machinery, a former Associate Editor for the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and a current Associate Editor for IEEE INTELLIGENT SYSTEMS.