Mach Learn (2007) 68: 171–200
DOI 10.1007/s10994-007-5015-9

Structured large margin machines: sensitive to data distributions

Daniel S. Yeung · Defeng Wang · Wing W.Y. Ng · Eric C.C. Tsang · Xizhao Wang

Received: 3 February 2005 / Revised: 17 January 2007 / Accepted: 21 April 2007 / Published online: 13 July 2007

Abstract This paper proposes a new large margin classifier—the structured large margin machine (SLMM)—that is sensitive to the structure of the data distribution. The SLMM approach incorporates the merits of "structured" learning models, such as radial basis function networks and Gaussian mixture models, with the advantages of "unstructured" large margin learning schemes, such as support vector machines and maxi-min margin machines. We derive the SLMM model from the concepts of "structured degree" and "homospace", based on an analysis of existing structured and unstructured learning models. Then, by using Ward's agglomerative hierarchical clustering on input data (or data mappings in the kernel space) to extract the underlying data structure, we formulate SLMM training as a sequential second order cone programming. Many promising features of the SLMM approach are illustrated, including its accuracy, scalability, extensibility, and noise tolerance. We also demonstrate the theoretical importance of the SLMM model by showing that it generalizes existing approaches, such as SVMs and M^4s, provides novel insight into learning models, and lays a foundation for conceiving other "structured" classifiers.

Editor: Dale Schuurmans.

This work was supported by the Hong Kong Research Grant Council under Grants G-T891 and B-Q519.

D.S. Yeung · D. Wang (✉) · W.W.Y. Ng · E.C.C. Tsang
Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
e-mail: csdfwang@comp.polyu.edu.hk

D.S. Yeung
e-mail: csdaniel@comp.polyu.edu.hk

W.W.Y. Ng
e-mail: cswyng@comp.polyu.edu.hk

E.C.C. Tsang
e-mail: csetsang@comp.polyu.edu.hk

X. Wang
Faculty of Mathematics and Computer Science, Hebei University, Baoding 071002, China
e-mail: wangxz@mail.hbu.edu.cn

Keywords Large margin learning · Weighted Mahalanobis distance (WMD) · Homospace · Structured learning · Agglomerative hierarchical clustering · Second order cone programming (SOCP)

1 Introduction

The supervised binary-learning problem is to differentiate between samples from two classes based on a set of features. To find a decision hyperplane $f(\mathbf{x}) : \mathbb{R}^n \to \{+1, -1\}$, which correctly predicts the class label of an unseen pattern, we train the classifier with m pairs of samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \mathbb{R}^n$ is an input vector labeled by $y_i \in \{+1, -1\}$. For many applications, such as network intrusion detection (Mukherjee et al. 1994), disease diagnosis (Christodoulou and Pattichis 1999), handwritten character recognition (Veeramachaneni and Nagy 2005), and human face detection (Osuna et al. 1997), data do appear in homogeneous groups. If the data structural information can be appropriately utilized to facilitate the classifier training, there will be a significant classification accuracy improvement.

Large margin classifiers (Smola et al. 2000) are popular approaches to solve the supervised learning problems. Founded upon Vapnik's statistical learning theory, the support vector machine (SVM) (Vapnik 1999; Burges 1998) has played an important role in many areas including (but not confined to) pattern recognition, regression, image processing, and bioinformatics, due to its salient properties such as margin maximization and kernel substitution for classifying the data in a high dimensional kernel space. However, as the SVM relies exclusively on a small number of "support vectors" to construct the decision hyperplane, which is blind to data distribution, there is still space for further improvement.

Linear discriminant analysis (LDA) (Fisher 1936) is another classification approach that employs the class structure to determine the decision boundary. Alternatives to LDA include the recently proposed minimax probability machine (MPM) (Lanckriet et al. 2002) and its extension—the minimum error minimax probability machine (MEMPM) (Huang et al. 2004b). Inspired by the underlying rationale of these methods, Huang et al. (2004a) proposed another large margin learning model, the maxi-min margin machine (M^4), that improves the SVM by considering class structures into decision boundary calculation via utilizing Mahalanobis distance as the distance metric. The M^4 does show better performance than the classical SVM in some applications, but as it just differentiates between classes, sometimes it is not as good as the SVM.

In contrast to the previous learning approaches, the radial basis function network (RBFN) (Haykin 1999) and the Gaussian mixture model (GMM) (Duda et al. 2001) divide data points into clusters and base their classification on how the samples distribute. Compared to the multilayer perceptron (MLP) (Haykin 1999), the RBFN usually achieves better overall performance, partially because of its proper consideration of data distribution by prior clustering.

The most significant difference observed from the above learning approaches is the granularity they "structure" the training data, i.e., the smallest unit where the data are considered to share the same distribution (usually measured by the covariance matrix). We name the homogeneous scope as the homospace. From this point of view, as the SVM is not sensitive to data distribution, its homospace is the individual data point. The M^4 structures training data into classes, thus its homospace is class. Therefore, both of them are unable to characterize the data trends, as illustrated in Fig. 1. For the star-shaped data points, they tend to spread in the direction perpendicular to the decision boundary, while the circular data points on the

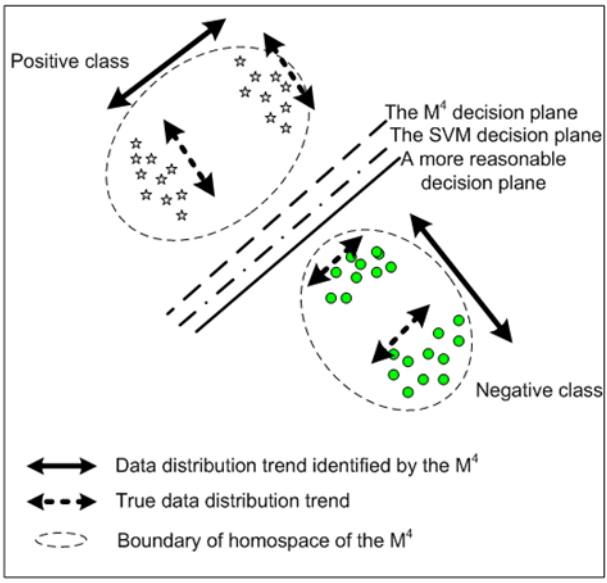


Fig. 1 Geometric interpretation of the homospace of the SVM and the M^4, and the decision boundaries calculated by the SVM, the M^4, and a desired classifier

other side tend to scatter in the orientation parallel to the decision plane. But the data distribution orientations detected by M^4s are nearly orthogonal to the above described directions, which totally misses the true data structure and misleads the classifier construction.

Considering these facts, we propose a large margin classifier with proper consideration of data structures. We call it the structured large margin machine (SLMM). The homospace of the SLMM is exactly the same as the data cluster. The data structure is quantified in two steps: (1) cluster the data points by Ward's linkage agglomerative hierarchical clustering; and (2) calculate the covariance matrix of each cluster for further distance measurement. The distance metric we use is called the weighted Mahalanobis distance (WMD), where the weight of each training pattern is determined by the size of the cluster it belongs to.

In Sect. 2, we will first define the homospace and the structured degree of a classifier, then use them to study several popular learning models. In Sect. 3, we present the formulation of the SLMM model, and elaborate on how it generalizes the SVM and the M^4. Experiments on toy and real-world data are given in Sect. 4 to empirically support our model. Sect. 5 mainly discusses the properties of the SLMM, and Section 6 gives the conclusion with possible directions for future work.

2 Structured learning and unstructured learning

2.1 Homospace

Definition 1 Suppose a training set T can be divided into n partitions, i.e., $H_1, \ldots, H_n$, where $H_1 \cup \cdots \cup H_n = T$, $H_i \cap H_j = \phi$, $i, j = 1, \ldots, n$ and $i \neq j$. If data points in each partition H_i are considered by the classifier C to share the same distribution trend, which empirically can be measured by the covariance matrix, these partitions, i.e., $H_1, \ldots, H_n$, are called homospaces of the classifier C for the training set T.

Therefore, the homospace is a classifier-specific concept, which characterizes "in what scope a certain classifier distinguishes data". One extreme is the SVM, not differentiating

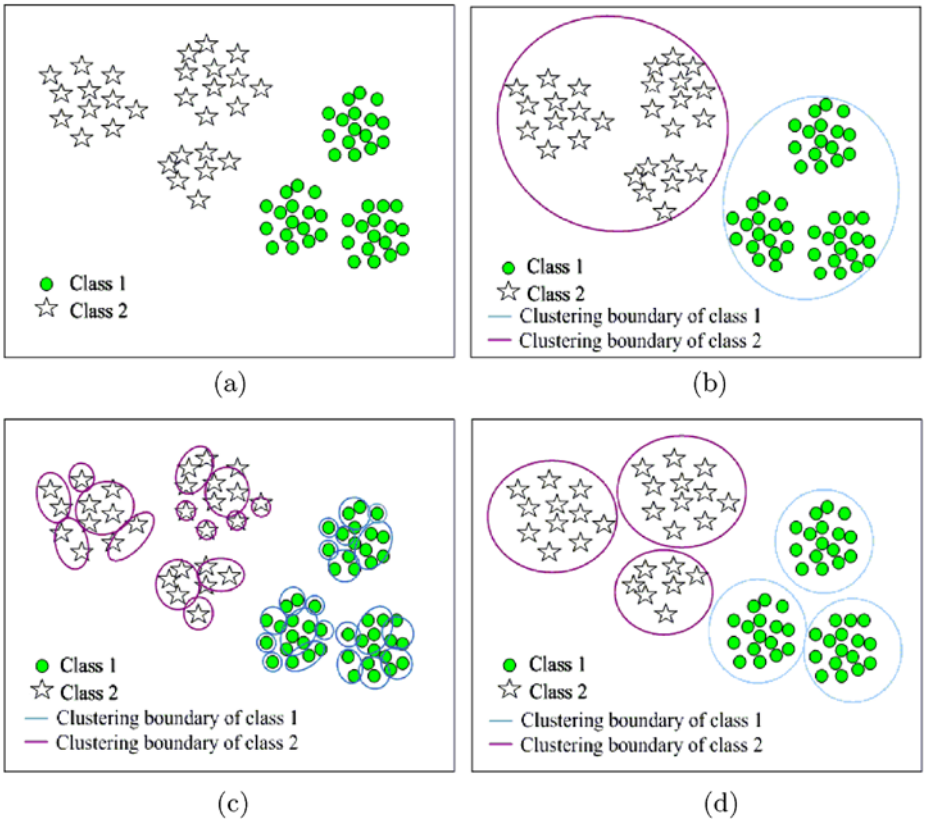


Fig. 2 Different structuring strategies on the same dataset: (**a**) structure as discrete points; (**b**) structure as 2 clusters; (**c**) structure as 35 clusters; (**d**) structure adapting to the data

data at all, with the homospace as small as an individual data point, while the other extreme is the M^4, assuming that data have the same distribution within each class, so its homospace is class. In this sense, the homospace can be regarded as an intrinsic property of the classifier.

2.2 Structured degree

We use the structured degree to describe the ability of a specific classifier to distinguish data structures. Suppose we have a classifier, which tries to find the data distribution information by clustering before classification. In Fig. 2(a), no clustering is performed, which is similar to that in SVMs. If we impose the number of clusters to be two, each corresponding to one class (cf. Fig. 2(b)), we approach the situation in LDAs, MPMs, and M^4s. Both of the above two cases are under-structuring and deserve low structured degree. On the other hand, over-structuring is also undesirable and will reduce the structured degree (cf. Fig. 2(c)). It is therefore preferred that the number of clusters is not fixed but adapts to the data (cf. Fig. 2(d)).

Definition 2 Suppose the real data distribution trends for training set T with dimension d are represented by the covariance information, i.e., $\mathbf{\Sigma}_{P_1}, \ldots, \mathbf{\Sigma}_{P_m}$, of m disjoint partitions

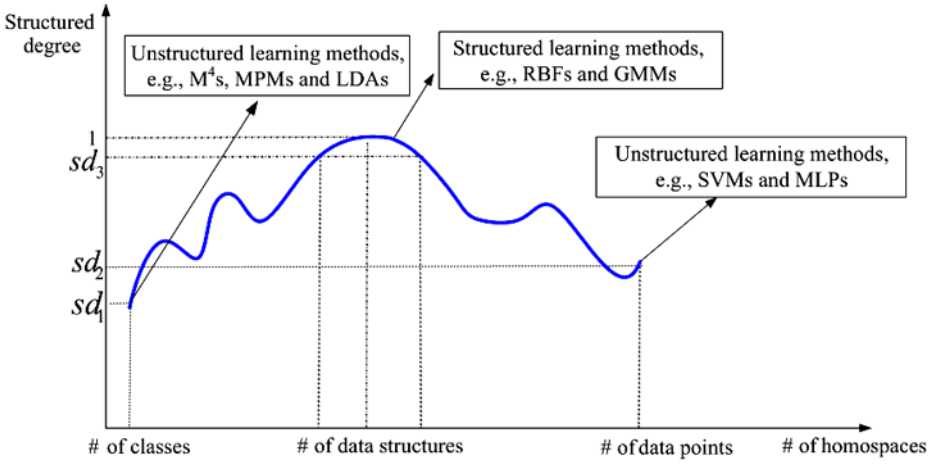


Fig. 3 The number of homospaces vs. the structured degree. The structured degree of "structured" learning methods (e.g., sd_3) is higher than those of "unstructured" learning methods (e.g., sd_1 and sd_2)

of T, namely $P_1, \ldots, P_m$. Assume the homospaces of C for T are $H_1, \ldots, H_n$ with covariance matrices $\mathbf{\Sigma}_{H_1}, \ldots, \mathbf{\Sigma}_{H_n}$, and ϵ is a small positive number. The structured degree (sd) of classifier C for training set T can be estimated by

$$sd_C(T) = \frac{\sum_{\mathbf{x}_i \in T}(1 - \frac{1}{d^2}\sum_{s=1}^{d}\sum_{t=1}^{d} D_{\mathbf{x}_i}(s,t))}{|T|},$$

where

$$D_{\mathbf{x}_i}(s,t) = \begin{cases} 1 & \text{if } |\sum_{P_k}(s,t) - \sum_{H_\ell}(s,t)| > \epsilon \text{ and } \mathbf{x}_i \in P_k \cap H_\ell, \\ 0 & \text{if } |\sum_{P_k}(s,t) - \sum_{H_\ell}(s,t)| \leq \epsilon \text{ and } \mathbf{x}_i \in P_k \cap H_\ell. \end{cases}$$

Applying the above definition of structured degree to analyze the existing learning models, we can draw the following conclusions:

- For a classifier with the homospace as a discrete data point, e.g., the SVM and the MLP, the covariance matrix $\mathbf{\Sigma}_{H_\ell}$ of the homospace H_ℓ is a zero matrix; thus, most elements in $D_{\boldsymbol{x}_i}$ are of the value 1, which makes its structured degree approach to 0.
- If the homospace of a classifier is class, e.g., the MPM, $\mathbf{\Sigma}_{H_\ell}$ is calculated from all the samples within a class, while $\mathbf{\Sigma}_{P_k}$ is computed from compact globular clusters. Thus the difference between $\mathbf{\Sigma}_{P_k}$ and $\mathbf{\Sigma}_{H_\ell}$ is large, and the structured degree will be small.
- It is reasonable to expect that if the homospace of a classifier is consistent with the unit that data points are inherently structured, the structured degree will mount up to the highest possible value, i.e., 1. With a proper clustering technique applied, the difference between $\mathbf{\Sigma}_{P_k}$ and $\mathbf{\Sigma}_{H_\ell}$ is likely to be smaller than ϵ and the structured degree is supposed to be high. The curve in Fig. 3 qualitatively illustrates the relationship between the number of homospaces and the structured degree, providing the same input data. The structured degree of "structured" learning methods (e.g., sd_3) is higher than those of "unstructured" learning methods (e.g., sd_1 and sd_2).

2.3 Structured learning vs. unstructured learning

Structured learning methods, e.g., RBFNs and GMMs, take the data structure into consideration when the classifier is being built. For RBFNs, clustering techniques give some heuristics to decide the topology of the network (the number of hidden neurons), with each cluster corresponding to one hidden neuron in RBFNs. GMMs estimate the *probability distribution function* for different classes separately via a mixture of several Gaussian distributions. Based on the Bayes theory, a test point is assigned to the class with the maximal posterior probability. If each class has equal prior probability, the test point is simply classified into the class that maximizes the probability distribution function. GMMs with the maximal likelihood have the following features: (1) data points in each Gaussian component are tight; and (2) all the data points are well covered in the components distribution. The above two properties are consistent with those of natural data clusters, which makes the GMM a "structured" learning approach.

There is another family of classifiers, e.g., MLPs and SVMs, which belongs to the unstructured learning category. The MLP trains the classifier using the back propagation (BP) algorithm, which aims to minimize the mean square error (MSE) between the desired output and the actual output. As the training process ignores how data points are distributed, it implicitly assumes that each data point is a homospace. In SVMs, the classification problem is transformed to a quadratic programming problem. Since there is no data distribution information contained both in the objective function and in the constraint inequalities, each data point is a homospace. Even though the M^4 model tries to capture the data trend of each class, it implicitly assumes that the data points inside each class share the same distribution. But in real-world applications, this assumption does not always hold. In some cases (cf. Fig. 1), the M^4 detects totally misleading structure from the data. Models like LDAs and MPMs can be seen as special cases of M^4s, thus they belong to the unstructured learning category as well.

For learning models with similar principles, such as mean square error minimizing approaches (e.g., the MLP and the RBFN), the "structured" one (the RBFN) outperforms the "unstructured" one (the MLP). Although the large margin learning methods have been widely applied, to the best of our knowledge, no work has been done to explore their "structured" counterparts. This is our major motivation to design a "structured" large margin learning model, called the structured large margin machine (SLMM). In the SLMM, we divide the data into several near-globular clusters. The margin is measured in the WMD and maximized via solving an optimization problem. The WMD from a training sample $\mathbf{x}_i$ to the decision boundary is calculated by considering both the size of the cluster to which $\mathbf{x}_i$ belongs and the Mahalanobis distance from $\mathbf{x}_i$ to the separating hyperplane. We give our taxonomy of learning models in Fig. 4, where the relationship of SLMMs with other learning models is clearly described.

3 Structured large margin machines (SLMMs)

The binary classification problem is the main focus of this paper. Given a dataset (generated independent identically distributed within each class) containing data points belonging to $P \subset \mathbb{R}^n$ (positive class) or $N \subset \mathbb{R}^n$ (negative class) with the target output $+1$ and -1, respectively, the problem is to find a decision hyperplane $f(\mathbf{x}) : \mathbb{R}^n \to \{+1, -1\}$ to separate these two classes with the greatest robustness. The test data point $\mathbf{x}$ will be classified to the class P if $f(\mathbf{x}) > 0$, otherwise to the class N. For clarity, Table 1 lists the notations that will be quoted in this section. The bold typeface denotes vectors and matrices, and the normal typeface stands for scalars and vector components.

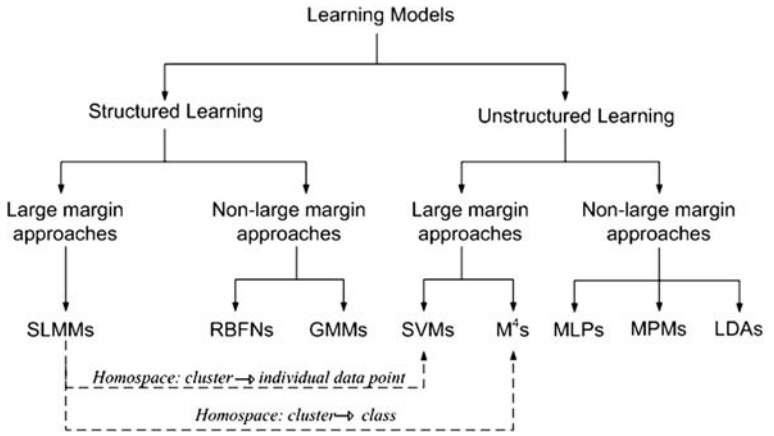


Fig. 4 A taxonomy of learning models. The relationship of SLMMs with other learning models (e.g., SVMs and M^4s) is clearly described

Table 1 Notation conventions used in this section

$\mathbf{x}_\ell, y_\ell$	the ℓ^{th} training pattern and its label
$\mathbf{x}, n$	input space, $n = \dim(\mathbf{x})$
$\mathbf{w}$	normal vector of a hyperplane
b	bias of a hyperplane
f	kernel space
$K(\cdot,\cdot)$	dot product in the kernel space
Φ	mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^f$
ξ_ℓ	"slack-variable" for data point $\mathbf{x}_\ell$
$\mathbf{\Sigma}_S$	covariance matrix of set S
$\lvert S\rvert$	size of set S
$\max\{S\}$	maximal value in set S

3.1 Methodologies

We first present linear SLMMs to solve linearly separable and nonseparable problems, which separates data points with a linear decision hyperplane in the input space. Then it is extended to a nonlinear version by using the kernel trick.

3.1.1 Linear SLMMs

I. The linearly separable case (*hard margin SLMMs*)
Assume the samples are linearly separable, i.e., there exist a vector $\mathbf{w}$ and a bias b satisfying the following constraints,

$$\begin{cases} \mathbf{x}_\ell^T \mathbf{w} + b > 0, & \text{if } \mathbf{x}_\ell \in P, \\ \mathbf{x}_\ell^T \mathbf{w} + b < 0, & \text{if } \mathbf{x}_\ell \in N. \end{cases}$$

Suppose there are C_P clusters in class P and C_N clusters in class N, i.e., $P = P_1 \cup \cdots P_i \cup \cdots P_{C_P}$, $N = N_1 \cup \cdots N_j \cup \cdots N_{C_N}$. Then the SLMM model can be formulated as

$$\max \quad \rho \tag{1a}$$

$$\text{s.t.} \quad (\mathbf{w}^T \mathbf{x}_\ell + b) \geq \frac{|P_i|}{\text{Max}_P} \rho \sqrt{\mathbf{w}^T \boldsymbol{\Sigma}_{P_i} \mathbf{w}}, \quad \mathbf{x}_\ell \in P_i, \tag{1b}$$

$$-(\mathbf{w}^T \mathbf{x}_\ell + b) \geq \frac{|N_j|}{Max_N} \rho \sqrt{\mathbf{w}^T \boldsymbol{\Sigma}_{N_j} \mathbf{w}}, \quad \mathbf{x}_\ell \in N_j, \tag{1c}$$

$$\mathbf{w}^T \mathbf{r} = 1, \tag{1d}$$

in which ρ is the margin, $Max_P = \max\{|P_1|, \ldots, |P_i|, \ldots, |P_{C_P}|\}$, $1 \leq i \leq C_P$, and $Max_N = \max\{|N_1|, \ldots, |N_j|, \ldots, |N_{C_N}|\}$, $1 \leq j \leq C_N$. $\mathbf{r}$ is a constant vector to limit the scale of the weight $\mathbf{w}$. Each element in $\mathbf{r}$ can be a random value that is non-zero. In fact, what matters is not the norm but the direction of $\mathbf{w}$. Appending the constraint $\mathbf{w}^T \mathbf{r} = 1$ makes the optimization problem solvable if $\mathbf{r}$ does not happen to be orthogonal to $\mathbf{w}$. We use the weighted Mahalanobis distance as our distance metric. $\boldsymbol{\Sigma}_{P_i}$ and $\boldsymbol{\Sigma}_{N_j}$ are the covariance matrices used to calculate the Mahalanobis distances $\frac{(\mathbf{w}^T \mathbf{x}_\ell + b)}{\sqrt{\mathbf{w}^T \boldsymbol{\Sigma}_{P_i} \mathbf{w}}}$ and $\frac{(\mathbf{w}^T \mathbf{x}_\ell + b)}{\sqrt{\mathbf{w}^T \boldsymbol{\Sigma}_{N_j} \mathbf{w}}}$ from data point $\mathbf{x}_\ell$ to the decision hyperplane. $\frac{|P_i|}{\text{Max}_P}$ and $\frac{|N_j|}{\text{Max}_N}$ are the weights indicating the importance of clusters P_i and N_j, respectively. If the cluster size is too small, the covariance matrix is assigned to be the identity matrix. Consequently, the data structure in terms of the covariance matrix of each cluster is taken into determining the decision hyperplane.

II. The linearly nonseparable case (*soft margin SLMMs*)
In cases where the samples are not linearly separable, the SLMM model is still workable by introducing the slack variable, ξ_ℓ. The optimization problem becomes

$$\max \quad \rho - C \sum_{\ell=1}^{|P|+|N|} \xi_\ell$$

$$\text{s.t.} \quad (\mathbf{w}^T \mathbf{x}_\ell + b) \geq \frac{|P_i|}{\text{Max}_P} \rho \sqrt{\mathbf{w}^T \boldsymbol{\Sigma}_{P_i} \mathbf{w}} - \xi_\ell, \quad \mathbf{x}_\ell \in P_i,$$

$$-(\mathbf{w}^T \mathbf{x}_\ell + b) \geq \frac{|N_j|}{\text{Max}_N} \rho \sqrt{\mathbf{w}^T \boldsymbol{\Sigma}_{N_j} \mathbf{w}} - \xi_\ell, \quad \mathbf{x}_\ell \in N_j,$$

$$\mathbf{w}^T \mathbf{r} = 1,$$

$$\xi_\ell \geq 0,$$

where $i = 1, \ldots, C_P$, $j = 1, \ldots, C_N$, and $\ell = 1, \ldots, |P| + |N|$. C is a constant denoting the tradeoff between the margin width ρ and the conceptually empirical error $\sum_{\ell=1}^{|P|+|N|} \xi_\ell$. This optimization problem can be interpreted as maximizing the WMD margin while minimizing the total training error.

3.1.2 Nonlinear SLMMs

According to Cover's pattern separability theory, patterns linearly nonseparable in the input space may be transformed into a kernel space to make them linearly separable, as long as

the transformation is nonlinear and the dimensionality of the kernel space is high enough (Haykin 1999). This nonlinear transformation can be achieved by using the Mercer kernel (Vapnik 1999; Schölkopf et al. 1999). Furthermore, the data topographical correlation in the input space will be preserved in the kernel space, if the nonlinear transformation is smooth and continuous (Girolami 2002). As linearly nonseparable patterns have the potential to be easily separated in the kernel space, the nonlinear SLMM using the kernel trick is developed to solve complex pattern classification tasks.

We first map the classification problem from the input space to the kernel space via a mapping function $\Phi\colon \mathbf{x}_\ell \to \Phi(\mathbf{x}_\ell)$, $\ell = 1, \ldots, |P| + |N|$, then derive the linear decision boundary $\mathbf{w}^T\Phi(\mathbf{x}) + b = 0$, where $\mathbf{w} \in \mathbb{R}^f$, $\Phi(\mathbf{x}) \in \mathbb{R}^f$, and $b \in \mathbb{R}$, in the kernel space, which actually corresponds to a nonlinear decision boundary in the original space. Thus, the optimization problem of the SLMM model in the kernel space can generally be formulated as follows

$$\max \quad \rho - C \sum_{\ell=1}^{|P|+|N|} \xi_\ell \tag{2a}$$

$$\text{s.t.} \quad (\mathbf{w}^T\Phi(\mathbf{x}_\ell) + b) \geq \frac{|P_i|}{\mathrm{Max}_P}\rho\sqrt{\mathbf{w}^T\boldsymbol{\Sigma}_{P_i}^{\Phi}\mathbf{w}} - \xi_\ell, \quad \mathbf{x}_\ell \in P_i, \tag{2b}$$

$$-(\mathbf{w}^T\Phi(\mathbf{x}_\ell) + b) \geq \frac{|N_j|}{\mathrm{Max}_N}\rho\sqrt{\mathbf{w}^T\boldsymbol{\Sigma}_{N_j}^{\Phi}\mathbf{w}} - \xi_\ell, \quad \mathbf{x}_\ell \in N_j, \tag{2c}$$

$$\xi_\ell \geq 0. \tag{2d}$$

For nonlinear SLMMs, we perform clustering in the kernel space. Suppose there are C_P clusters in the positive class and C_N in the negative class, $\mathrm{Max}_P = \max\{|P_i|\}, 1 \leq i \leq C_P$, $\mathrm{Max}_N = \max\{|N_j|\}, 1 \leq j \leq C_N$. The optimization problem described in (2a–2d) is not solvable unless it is represented in the kernel form $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$, i.e., a dot product of maps of samples.

Theorem 1 *If the estimates of mean and covariance matrix of cluster C in the kernel space are respectively*

$$\boldsymbol{\mu}_C^{\Phi} = \frac{1}{|C|}\sum_{\mathbf{x}\in C}\Phi(\mathbf{x})$$

and

$$\boldsymbol{\Sigma}_C^{\Phi} = \frac{1}{|C|}\sum_{\mathbf{x}\in C}(\Phi(\mathbf{x}) - \mathbf{m}_C^{\Phi})(\Phi(\mathbf{x}) - \mathbf{m}_C^{\Phi})^T,$$

the optimal $\mathbf{w}$ *in the optimization problem* (2a–2d) *lies in the space spanned by the training data maps* (*cf. Appendix* 1 *for the proof*).

According to Theorem 1, we can write $\mathbf{w}$ as $\sum_{i=1}^{|P|+|N|}\alpha_i\Phi(\mathbf{x}_i)$, where $\alpha_i \in \mathbb{R}$ are coefficients. By simply substituting $\mathbf{w}$ into the optimization problem (2a–2d), we can obtain the kernel form of the optimization problem (cf. Appendix 2),

$$\max \quad \rho - C \sum_{\ell=1}^{|P|+|N|} \xi_\ell \tag{3a}$$

$$\text{s.t.} \quad (\mathbf{K}_\ell \boldsymbol{\alpha} + b) \geq \frac{|P_i|}{\text{Max}_P} \rho \sqrt{\boldsymbol{\alpha}^T \tilde{\mathbf{K}}_{P_i}^T \tilde{\mathbf{K}}_{P_i} \boldsymbol{\alpha}} - \xi_\ell, \quad \mathbf{x}_\ell \in P_i, \tag{3b}$$

$$-(\mathbf{K}_\ell \boldsymbol{\alpha} + b) \geq \frac{|N_j|}{\text{Max}_N} \rho \sqrt{\boldsymbol{\alpha}^T \tilde{\mathbf{K}}_{N_j}^T \tilde{\mathbf{K}}_{N_j} \boldsymbol{\alpha}} - \xi_\ell, \quad \mathbf{x}_\ell \in N_j, \tag{3c}$$

$$\boldsymbol{\alpha}^T \mathbf{r} = 1, \tag{3d}$$

$$\xi_\ell \geq 0. \tag{3e}$$

Similar to (1d), (3d) is used to constrain the magnitude of $\boldsymbol{\alpha}$ and the constant non-zero vector $\mathbf{r}$ is required not to be orthogonal to $\boldsymbol{\alpha}$. $\mathbf{K}_\ell$ represents the ℓth row in the kernel Gram matrix $\mathbf{K}$, in which the elements satisfy $\mathbf{K}(i, j) = K(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, |P| + |N|$. $\tilde{\mathbf{K}}_{P_i} = \frac{1}{\sqrt{|P_i|}}(\mathbf{K}_{P_i} - \mathbf{e}_{|P_i|} \cdot \mathbf{v}_{P_i}^T)$, where $\mathbf{e}_{|P_i|}$ is an all-one column vector with length $|P_i|$. $\mathbf{K}_{P_i}$ is the kernel matrix between the cluster P_i and all the training patterns $\mathbf{K}_{P_i}(s, j) = K(\mathbf{x}_s, \mathbf{x}_j)$, $s = 1, \ldots, |P_i|$, $j = 1, \ldots, |P| + |N|$. $\mathbf{v}_{P_i}$ is the mean vector of matrix $\mathbf{K}_{P_i}$ and $\mathbf{v}_{P_i}(j) = \sum_{\mathbf{x}_s \in P_i} K(\mathbf{x}_s, \mathbf{x}_j)/|P_i|$, $j = 1, \ldots, |P| + |N|$. $\tilde{\mathbf{K}}_{N_j}$ is calculated similar to $\tilde{\mathbf{K}}_{P_i}$.

3.2 Key issues

To establish the SLMM model, some key issues still deserve careful consideration. We discuss in this subsection two of them, i.e., the clustering technique and the optimization problem solving.

3.2.1 On the clustering method

For the purpose of investigating the structure of a given dataset, the hierarchical clustering (Jain and Dubes 1988) is adopted to detect the clusters in each individual class. For linear SLMMs, the input patterns are clustered hierarchically in the input space, while for nonlinear SLMMs, the hierarchical clustering is performed in the kernel space. Specifically, SLMMs cluster data points in an agglomerative manner, which can be formally described as follows.

Initialize each point as a cluster and calculate the distance between every two clusters
While more than one cluster remains
  Find the closest pair of clusters
  Merge the two clusters
  Update the distance between each pair of clusters
End

The output of this algorithm is a tree structure known as the dendrogram (Everitt et al. 2001), whose topology is also a representation of the clustering process. Therefore, by cutting this dendrogram at different levels, one can achieve diverse clustering results.

Various hierarchical clustering approaches (Jain and Dubes 1988), e.g., single linkage clustering, complete linkage clustering, centroid linkage clustering and Ward's linkage clustering, differ in the method of finding the closest pair of clusters. We use the Ward's linkage clustering (Ward 1963) in this study for the reason that clusters derived from this method are compact and spherical (El-Hamdouchi and Willett 1989), which provides a meaningful basis for the calculation of covariance matrices and therefore for the computation of (weighted) Mahalanobis distances. If S and T are two clusters with means $\boldsymbol{\mu}_S$ and $\boldsymbol{\mu}_T$, respectively, the

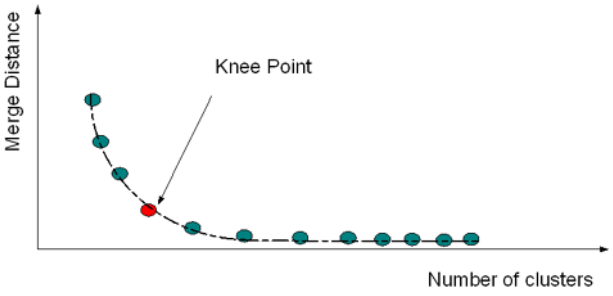


Fig. 5 Choosing the *knee point* as the optimal number of clusters. The *knee point* is the point of maximum curvature

Ward's linkage $W(S,T)$ between clusters S and T can be calculated as

$$W(S,T) = \frac{|S| \cdot |T| \cdot \|\boldsymbol{\mu}_S - \boldsymbol{\mu}_T\|^2}{|S| + |T|}. \tag{4}$$

Initially, each pattern is a cluster. The Ward's linkage of two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ is $W(\mathbf{x}_i, \mathbf{x}_j) = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2}$. When two clusters A and B are being merged to a new cluster A', to be more computationally efficient, $W(A', C)$ can be conveniently derived from $W(A, C)$, $W(B, C)$, and $W(A, B)$ by

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

During hierarchical clustering, the Ward's linkage between clusters to be merged increases as the number of clusters decreases. A curve, namely the merge distance curve, is drawn to represent this process. The dendrogram can be cut when given the number of clusters, which can be determined by finding the *knee point* (Salvador and Chan 2004), i.e., the point of maximum curvature, as shown in Fig. 5.

In the high-dimensional, implicit kernel space, the hierarchical clustering is still applicable:

1. The Ward's linkage between $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$, i.e., the images of patterns $\mathbf{x}_i$ and $\mathbf{x}_j$, can be calculated by $W(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) = \frac{1}{2}[K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j)]$ (cf. (4));
2. When two clusters A^Φ and B^Φ merge to a new cluster A'^Φ, the Ward's linkage between A'^Φ and C^Φ can be conveniently calculated by (cf. Appendix 3 for the derivation)

$$W(A'^\Phi, C^\Phi) = \frac{(|A^\Phi| + |C^\Phi|t)W(A^\Phi, C^\Phi) + (|B^\Phi| + |C^\Phi|)W(B^\Phi, C^\Phi) - |C^\Phi|W(A^\Phi, B^\Phi)}{|A^\Phi| + |B^\Phi| + |C^\Phi|}.$$

Complexity analysis In the initialization step, a Ward's linkage is calculated for each pair of patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ (or $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$) in the same class, thus the time complexity for this step is $O((|P|^2 + |N|^2) \cdot n)$. Without loss of generality, take the positive class for example, there are $|P| - 1$ rounds of merging, and the complexity for each is $O(m \cdot n)$, where m is the number of clusters that monotonically decreases but is always less than $|P|$; hence, the total complexity for these steps is $O((|P| - 1) \cdot m \cdot n)$. Therefore, the overall complexity for the agglomerative hierarchical clustering is $O((|P|^2 + |N|^2) \cdot n)$ in the input space or the kernel space.

3.2.2 On the optimization method

Since each constraint in the optimization problem of the SLMM model is either in the second order conic form or in the linear form, and the cost function is linear, the optimization problem for a specific ρ is a second order cone programming (SOCP) problem (Lobo et al. 1998), which can be handled efficiently by existing programs such as SeDuMi (Sturm 1999) or Mosek (Andersen and Andersen 2001). Therefore, the optimization problem in SLMMs can be solved via a line search and solving a sequential SOCP problem, similar to M^{4}s (Huang et al. 2004a).

Complexity analysis The time complexity of building the constraint matrix for the SOCP problem is $O((|P|+|N|)\cdot n^3)$. The worst case cost for solving each SOCP using the interior-point method is $O(n^3)$ (Lobo et al. 1998). Based on the chosen line search method, suppose t SOCP problems have to be solved to reach the required precision, the total complexity is $O((|P|+|N|)\cdot n^3 + t\cdot n^3) \approx O((|P|+|N|)\cdot n^3)$. This complexity is the same as solving the optimization problem in M^{4}s.

3.3 Relationship of SLMMs with M^{4}s and SVMs

In this subsection, our SLMM model is compared with two large margin classifiers, i.e., the SVM and the M^4. These analyses demonstrate how our model can be transformed to the SVM and the M^4 under some special conditions. With these supports, the SLMM can be viewed as the generalization of the SVM and the M^4. For simplicity but without loss of generality, we only analyze the linearly separable case.

3.3.1 Relationship with M^{4}s

If one assumes there exists only one cluster in each class, i.e., $C_P = C_N = 1$, and $\mathbf{\Sigma}_P$ ($\mathbf{\Sigma}_N$) is the covariance matrix for the positive (negative) class, the optimization problem (1a–1d) in SLMMs can be immediately converted to

$$\max \quad \rho \tag{5a}$$

$$\text{s.t.} \quad \frac{(\mathbf{w}^T\mathbf{x}_\ell + b)}{\sqrt{\mathbf{w}^T\mathbf{\Sigma}_P\mathbf{w}}} \geq \rho, \quad \mathbf{x}_\ell \in P, \tag{5b}$$

$$\frac{-(\mathbf{w}^T\mathbf{x}_\ell + b)}{\sqrt{\mathbf{w}^T\mathbf{\Sigma}_N\mathbf{w}}} \geq \rho, \quad \mathbf{x}_\ell \in N, \tag{5c}$$

$$\mathbf{w}^T\mathbf{r} = 1, \tag{5d}$$

where (5a–5c) is exactly of the same form as in M^{4}s (Huang et al. 2004a). The above optimization problem without constraint (5d) is proved not to converge (cf. Appendix 4 for the proof).

Geometrically speaking, the Mahalanobis distance used in M^{4}s only makes sense in distributions that look like nice globular clouds (Devroye et al. 1996). However, the training data are not always the case in real-world tasks (cf. Fig. 10). Moreover, by recalling the conceptual scheme we elaborated on in Sect. 3, its homospace is not consistent with the natural manner data agglomerate. In contrast, the SLMM approach divides the training data in one class into several globular clusters, so its homospace has a natural correlation with the real data groups, and this correlation is reflected in the distance metric WMD. As it has been demonstrated mathematically (Huang et al. 2004a) that MPMs and LDAs are special cases of M^{4}s, we can conveniently take them as special cases of SLMMs.

3.3.2 Relationship with SVMs

If one assumes each data point to be a cluster, i.e., $C_P = |P|$ and $C_N = |N|$, and $\mathbf{\Sigma}_{P_i} = \mathbf{\Sigma}_{N_j} = \mathbf{I}$ ($\mathbf{I}$ represents the identity matrix), $i = 1, \ldots, C_P$, $j = 1, \ldots, C_N$, the optimization problem (1a–1d) in SLMMs can be written as

$$\max \quad \rho \tag{6a}$$

$$\text{s.t.} \quad (\mathbf{w}^T \mathbf{x}_\ell + b) \geq \rho \|\mathbf{w}\|, \quad \mathbf{x}_\ell \in P, \tag{6b}$$

$$-(\mathbf{w}^T \mathbf{x}_\ell + b) \geq \rho \|\mathbf{w}\|, \quad \mathbf{x}_\ell \in N, \tag{6c}$$

$$\mathbf{w}^T \mathbf{r} = 1. \tag{6d}$$

The norm of $\mathbf{w}$ is not important; what really matters is its orientation. As constraint (6d) is to limit the magnitude of $\mathbf{w}$, another form of limitation on $\mathbf{w}$ is $\rho\|\mathbf{w}\| = 1$, which can replace (6d) to serve the same purpose. Therefore, optimization (6a–6d) can be formulated as

$$\min \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t.} \quad (\mathbf{w}^T \mathbf{x}_\ell + b) \geq 1, \quad \mathbf{x}_\ell \in P,$$

$$-(\mathbf{w}^T \mathbf{x}_\ell + b) \geq 1, \quad \mathbf{x}_\ell \in N,$$

which is exactly the optimization problem defined in classical SVMs (Vapnik 1999; Burges 1998), and the distance metric WMD herein degenerates to the Euclidean distance.

4 Experiments and results

4.1 On synthetic datasets

4.1.1 The linearly separable case (hard margin SLMMs)

Experiment I. Results and comparisons
The synthetic two-dimensional dataset is randomly generated under two Gaussian distributions for the positive or negative class (see Table 2 for the statistical values). The training set has 60 samples (30 samples in each class), and the testing set contains 60 points in each class. Samples in either class are designed to scatter in two clusters: N_1 (P_1) and N_2 (P_2)

Table 2 Linearly separable data generation: the size, mean, and covariance matrix of each cluster. Each class contains two clusters

		Probability	Mean	Covariance
Positive Class	Gaussian Distribution P_1	2/3	$[4, -5]^T$	[6, 0; 0, 0.5]
	Gaussian Distribution P_2	1/3	$[5, 5]^T$	[6, 0; 0, 0.5]
Negative Class	Gaussian Distribution N_1	2/3	$[-2, 0]^T$	[0.5, 0; 0, 6]
	Gaussian Distribution N_2	1/3	$[-11, -1]^T$	[0.5, 0; 0, 6]

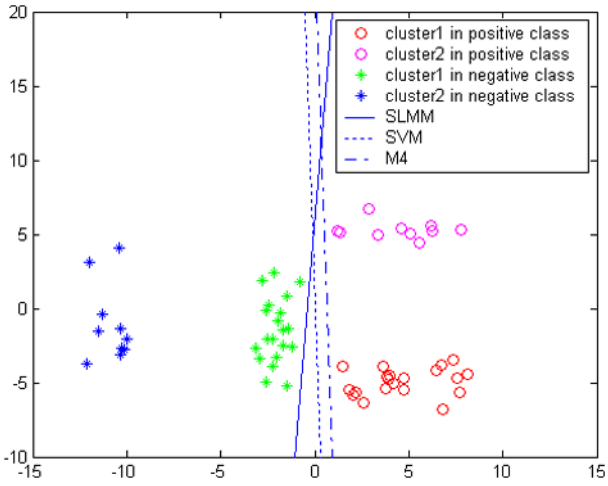


Fig. 6 Classification results of the SLMM, the SVM, and the M^4 on a linearly separable synthetic dataset. **a** Decision planes built on training data. **b** Classification of testing data with the resulting decision planes. The training accuracies for the SLMM, the SVM, and the M^4 are all 100%, but the testing accuracies for them are 97.5%, 95.8%, 93.3%, respectively

for the negative (positive) class. The testing accuracies are 97.5%, 95.8%, 93.3% for the SLMM, the SVM, and the M^4, respectively. The resulting decision boundaries are shown in Fig. 6. From the results, we have observations as follows:

- The SLMM achieves higher testing accuracy by considering the following data structures: clusters N_1 and N_2 scatter vertically, while clusters P_1 and P_2 spread horizontally, and P_1 is larger. The SLMM catches these structures and derives the separating hyperplane that leaves more room for P_1.
- The SVM ignores the data structures and obtains the boundary unbiasedly in the middle of the support vectors.

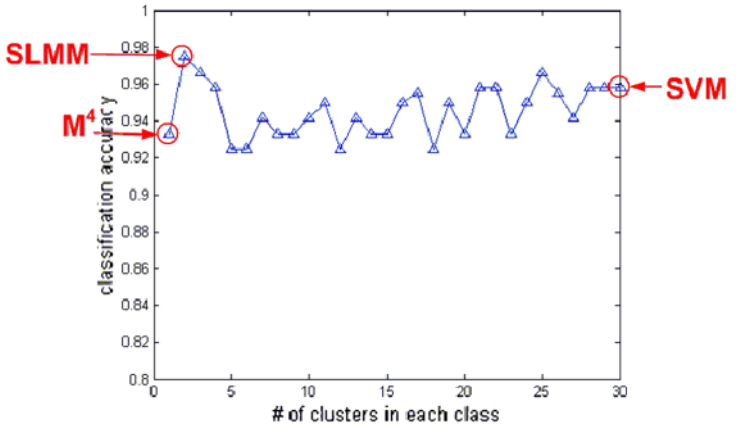


Fig. 7 Classification accuracies providing different numbers of clusters in each class. This fluctuating curve indicates that the number of clusters does affect the classification result

- In the M^4, as the covariance matrix is derived from each class, but not from globular clusters in the class, it generates a decision plane different from the one that is generated by the SLMM.

Experiment II. The influence of the number of clusters on the performance of SLMMs
In this experiment, we attempt to explore the relationship between the number of clusters and the performance of SLMMs, and to support our claim in Fig. 3. With the same experiment settings and data as in *Experiment I*, we impose the number of clusters c in each class to be an integer ranging from 1 to 30. If c is equal to 1, the whole class is considered as a single cluster, just as in the M^4. On the other hand, each pattern will become a cluster if c equals 30, the number of training patterns, which coincides with the SVM. With c set to different values, we record the classification accuracies of the SLMM (cf. Fig. 7). This fluctuating curve indicates that the number of clusters c does affect the classification result, and the value of c that corresponds to the highest accuracy is 2, which is consistent with the number of clusters determined by the *knee point* method. When the number of clusters is at its minimal or maximal value, corresponding to the situations in the M^4 and the SVM, respectively, the classification accuracy is not good enough due to their failure in capturing the true data distribution. For other assignments of clusters, the data distribution trends are more or less lost, thus the accuracy fluctuates.

4.1.2 The linearly nonseparable case (soft margin SLMMs)

This experiment aims to evaluate SLMMs and other models when data points are linearly nonseparable. The synthetic data for this experiment is generated with the statistics listed in Table 3. The classification results are illustrated in Fig. 8.

There are 60 training points (30 samples in each class), and 180 testing points (90 samples in each class). It is observable that the training points in N_2 and P_2 have some overlapping, which is more significant in the testing data (cf. Fig. 8). The overall data distribution is horizontal, but in the negative class, N_2 contains more data points than N_1, which implies a stronger tendency for the negative class to scatter in the lower area. Similarly, for the positive class, the data have a stronger tendency to spread in the upper area. The SLMM captures these distributions and determines the decision boundary that leaves more space for P_1 and

Table 3 Linearly nonseparable data generation: the size, mean, and covariance matrix of each cluster. Each class contains two clusters

		Probability	Mean	Covariance
Positive Class	Gaussian Distribution P_1	2/3	$[3.5, 5]^T$	[6, 0; 0, 0.5]
	Gaussian Distribution P_2	1/3	$[3.5, -5]^T$	[4.5, 0; 0, 0.5]
Negative Class	Gaussian Distribution N_1	1/3	$[-4, 5.5]^T$	[5, 0; 0, 0.5]
	Gaussian Distribution N_2	2/3	$[-4, -4.5]^T$	[7, 0; 0, 0.6]

Table 4 XOR data generation for the nonlinear SLMM: the size, mean, and covariance matrix of each cluster. Each class contains two clusters

		Probability	Mean	Covariance
Positive Class	Gaussian Distribution P_1	1/2	$[5.5, 0]^T$	[0.5, 0; 0, 6]
	Gaussian Distribution P_2	1/2	$[-5.5, 0]^T$	[8, 0; 0, 0.5]
Negative Class	Gaussian Distribution N_1	1/2	$[1, 8]^T$	[0.5, 0; 0, 5]
	Gaussian Distribution N_2	1/2	$[0, -5]^T$	[5, 0; 0, 0.5]

N_2. In comparison, both the SVM and the M^4 fail in sensing the data tendency. The training accuracies for the SLMM, the SVM and the M^4 are all 96.67%, but the testing accuracies for them are 95.56%, 92.78% and 93.33%, respectively. Note that the regularization parameter C is set to 20 for all three models.

4.1.3 Nonlinear SLMMs

In this experiment, we generate 80 (40 for each class) training samples, and 800 (400 for each class) testing samples. The data generation statistics are described in Table 4. For the SLMM, the SVM and the M^4, the regularization parameter C is 10 and the width parameter σ in the Gaussian kernel is 0.5. With these settings, all the three models are tested and their classification accuracies are 99.38%, 93.75%, and 95.00%, respectively. It can be concluded from the result, as shown in Fig. 9, that the SLMM detects the correct data tendency and reserves more space for N_1 in the vertical direction and for N_2 in the horizontal direction (cf. Fig. 9).

4.2 On real-world benchmark datasets

We test our proposed model and compare it with the SVM, the M^4, and the RBFN also on the benchmark datasets obtained from the UCI Machine Learning Repository:[1] IONOSPHERE (classification of radar returns from the ionosphere), PIMA (classification of diabetes in Pima Indians), SONAR (classification of sonar signals), HEART (diagnosis of the heart disease), BREAST (diagnosis of the breast cancer), and six datasets provided by Raetsch:[2] RINGNORM, IMAGE, SOLAR, TITANIC, GERMAN, and BANANA. For all the datasets, the number of training patterns, the number of testing patterns, and the dimension of the

[1] http://www.ics.uci.edu/∼mlearn/ML-Repository.html.

[2] http://mlg.anu.edu.au/∼raetsch/data/index.html.

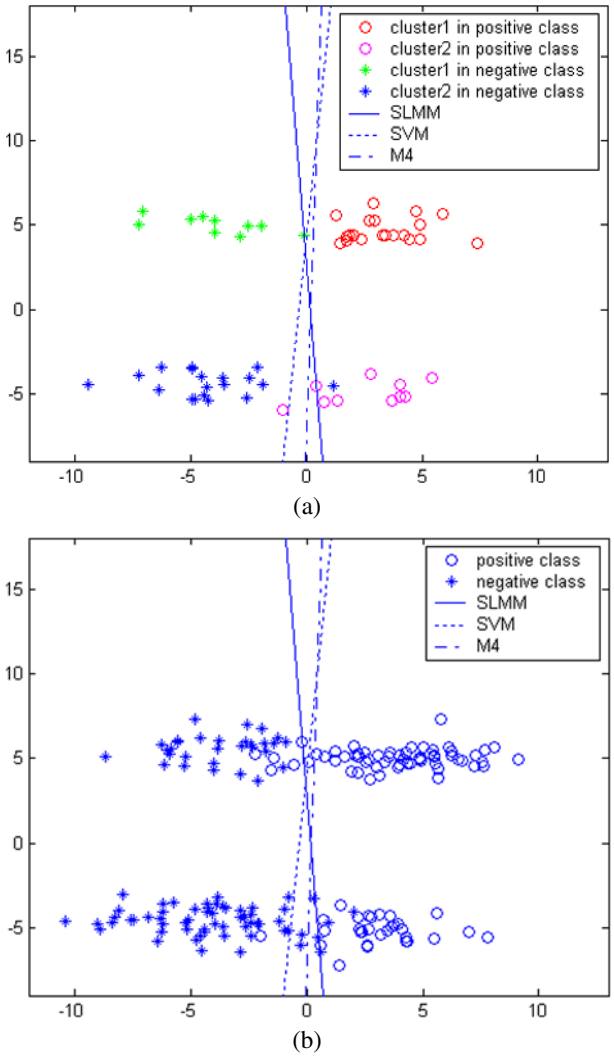


Fig. 8 Classification results of the SLMM, the SVM, and the M^4 models on a linearly nonseparable synthetic dataset. **a** Decision planes built on training data. **b** Classification of testing data with the resulting decision planes. The training accuracies for the SLMM, the SVM and the M^4 models are all 96.67%, but the testing accuracies are 95.56%, 92.78% and 93.33%, respectively

input data are characterized by the three numbers respectively following the name of each dataset in Table 5. We also list the average training and average testing time over all datasets for different methods.

The regularization parameter C and the width parameter σ in the Gaussian kernel, for SLMMs, M^4s, and SVMs, are tuned using 30-fold cross-validation. For RBFNs, we use the optimization algorithm as described in (Rätsch et al. 2001) to find the number of centers and their width values. The resulting accuracies shown in Table 5 are the averages over 100

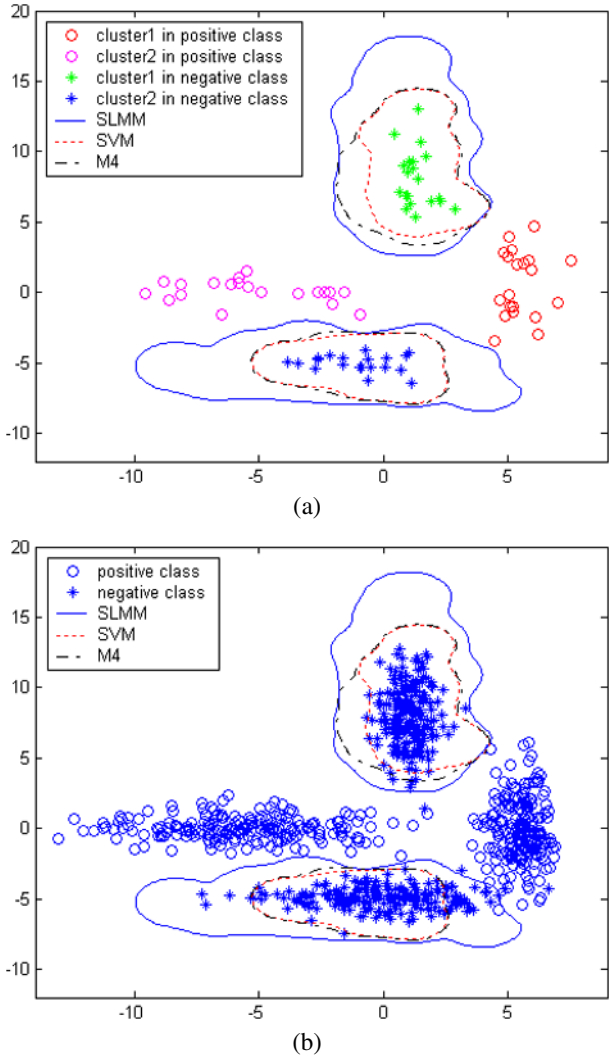


Fig. 9 Classification results of the SLMM, the SVM, and the M^4 on the XOR dataset. **a** Decision boundaries built on training data. **b** Classification of testing data with the resulting decision boundaries. The training accuracies for the SLMM, the SVM, and the M^4 are all 100%, but the testing accuracies are 99.38%, 93.75%, and 95.00%, respectively

random partitions of the data. We can draw the following conclusions from the experimental results.

- Compared with M^4s, SVMs, and RBFNs, SLMMs achieve the best overall performance due to the sensitivity to data distribution.
- M^4s are not always better than SVMs because the method used to extract data distribution information is misleading in some cases.
- Almost all datasets have more or less an intrinsic structure. Taking advantage of the distribution information is very likely to improve the classification accuracy.

Table 5 Comparison of experimental results on benchmark datasets

Database (#train × #test × dim.)	SLMM (%)	M^4 (%)	SVM (%)	RBFN (%)
BANANA (400 × 900 × 2)	91.4 ± 0.5	89.2 ± 1.1	88.4 ± 0.6	89.5 ± 0.3
BREAST (350 × 349 × 9)	97.4 ± 0.5	97.1 ± 0.6	96.8 ± 0.5	95.5 ± 0.7
GERMAN (700 × 300 × 20)	79.6 ± 2.0	76.2 ± 2.6	76.5 ± 2.1	75.3 ± 2.3
HEART (152 × 151 × 14)	87.1 ± 0.5	86.0 ± 0.8	84.2 ± 0.5	83.2 ± 1.0
IONOSPHERE (176 × 175 × 34)	94.6 ± 0.4	94.1 ± 0.5	93.9 ± 0.4	92.4 ± 0.8
IMAGE (1300 × 1010 × 18)	98.6 ± 0.4	96.2 ± 0.6	97.1 ± 0.5	96.7 ± 0.4
PIMA (384 × 384 × 8)	80.6 ± 0.5	77.6 ± 0.8	77.9 ± 0.5	75.9 ± 0.6
RINGNORM (400 × 7000 × 20)	98.3 ± 0.1	97.9 ± 0.3	98.4 ± 0.1	98.2 ± 0.2
SOLAR (666 × 400 × 9)	69.5 ± 1.4	66.9 ± 2.1	67.6 ± 1.6	65.6 ± 1.8
SONAR (104 × 104 × 60)	87.5 ± 0.8	84.8 ± 1.2	86.5 ± 1.1	83.9 ± 1.0
TITANIC (150 × 2051 × 3)	78.6 ± 1.0	77.9 ± 1.2	77.8 ± 1.1	76.5 ± 1.4
Average Training Time (Sec.)	49.7	43.6	16.0	26.9
Average Testing Time (Sec.)	7.9	8.2	7.4	4.8

- SLMMs and M^4s have similar training time, which is longer than that of SVMs. The testing time is almost equal for the three large margin models.

The SLMM outperforms the other three models mainly because of the proper consideration of the data structure information. In order to demonstrate the existence of data structure in the Gaussian kernel space, which is impractical to be directly displayed because of the infinite dimensionality, we choose to plot samples in the kernel space by kernel principal component analysis (KPCA) (Schölkopf et al. 1998), i.e., projecting them onto the three most principal kernel components in the kernel space. The structures of several datasets are illustrated in Fig. 10. Note that the values for the kernel widths are just the same as in SLMMs training.

5 Discussion

5.1 On the theoretical framework

Recently, Huang et al. (2004a) proposed a taxonomy, namely "global" learning vs. "local" learning, to categorize existing learning models. This viewpoint broadens the traditional scope that is concerned in most established learning machines. However, no criterion is explicitly mentioned to judge between "global" and "local" learning. Thus it is a blurry, conceptual taxonomy. Whereas, in this paper, we not only describe a system to classify the learning models, but also elaborate on the operable measures, i.e., the homospace and the structured degree. With these concepts, the influence of data distribution on classification for a certain classifier can be quantitatively characterized in some sense. In practice, we can perform this quantification on a dataset with known structure as the standard, so that the evaluation and comparison of classifiers in terms of the structured degree or the homospace become reasonable and fair.

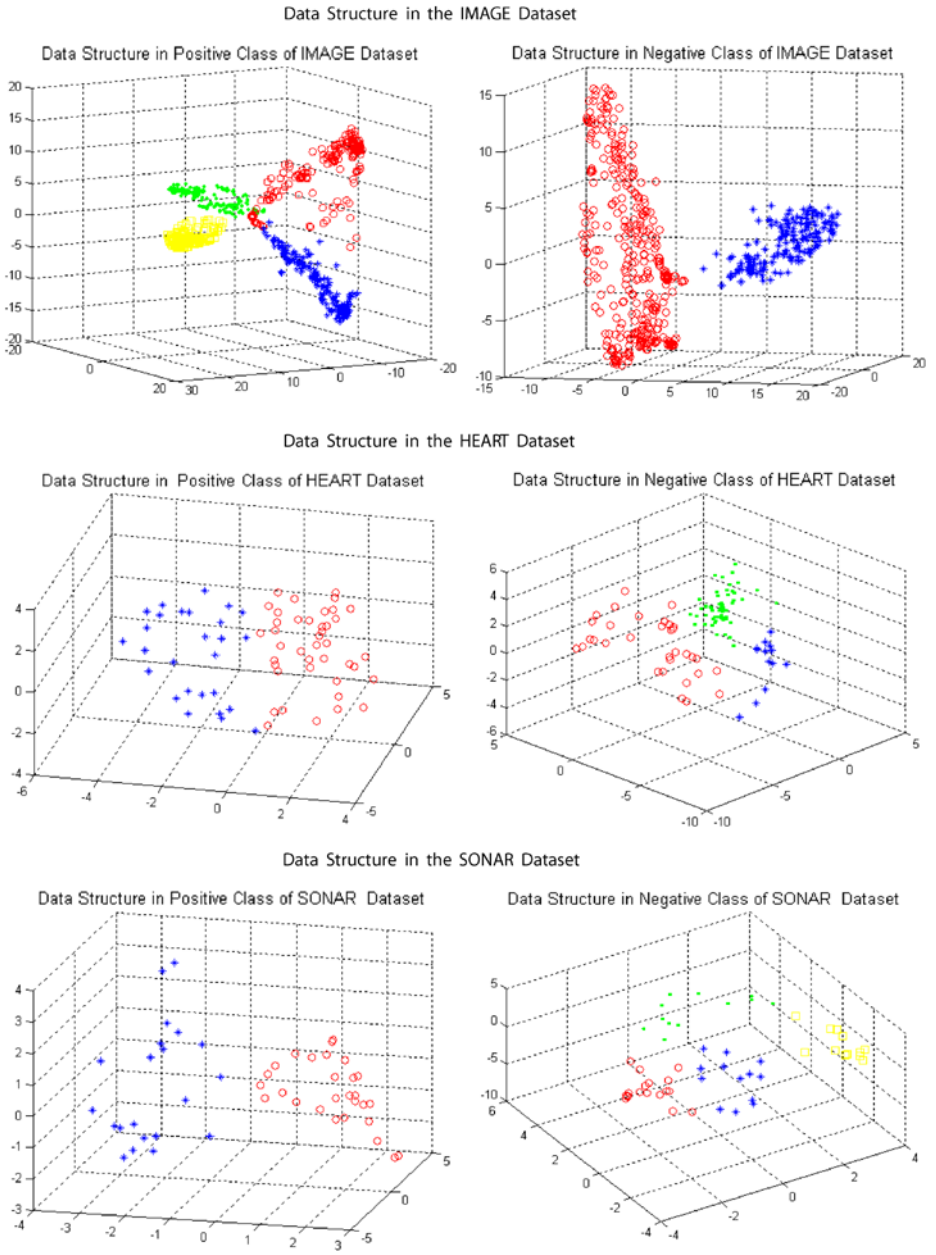


Fig. 10 Visualization of data structures found in both classes of datasets IMAGE, HEART, and SONAR by projecting the samples in the kernel space onto the three most principal kernel components

5.2 On the noise suppression capability

SLMMs also possess another attractive advantage, i.e., the powerful noise tolerance. That is to say, the resulting decision hyperplane is less sensitive to the existence of miscollected

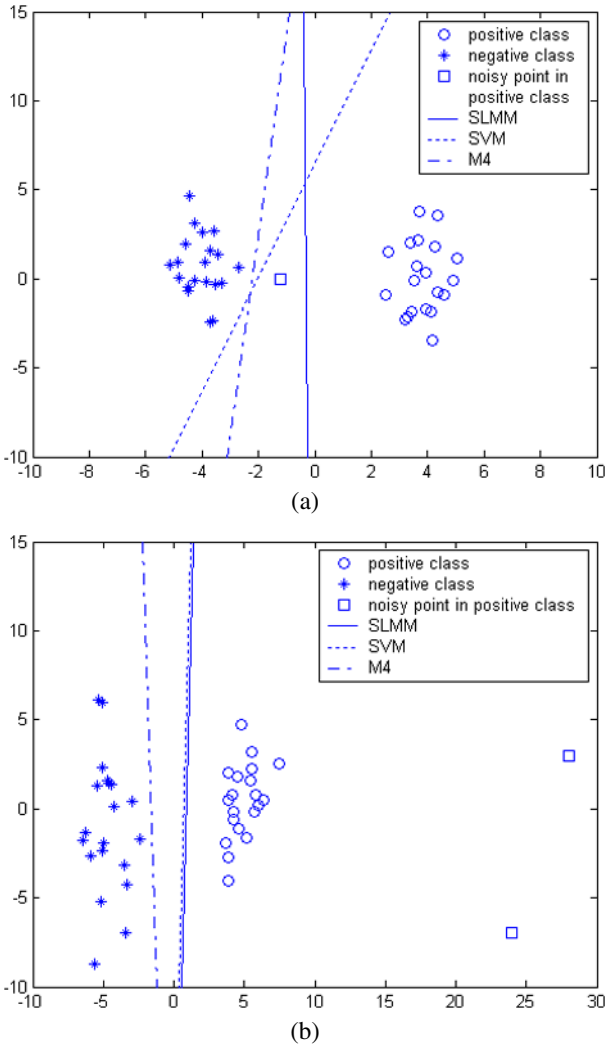


Fig. 11 Comparison of noise suppression abilities of the SLMM, the M^4, and the SVM. **a** In the case of near noises, the classification accuracies on testing data for the SLMM, the SVM, and the M^4 are 100%, 97.75%, and 99.25%, respectively. **b** In the case of distant noises, the test accuracies are 100%, 100%, and 99.25%, respectively

patterns when compared with other large margin machines. As illustrated in Fig. 11(a), when an isolated noisy point exists between two training classes, the SVM mistakenly treats it as one support vector, and determines the decision plane that is nearer to the negative class. The M^4 is not good at dealing with the problem in Fig. 11(a) either. Moreover, it is also sensitive to the isolated noises distant from the decision plane, because they cast a lot of impact upon the calculation of covariance of each class. Consequently, the resultant decision boundary of the M^4 will be misled as well (cf. Fig. 11(b)). However, the SLMM regards noisy patterns as tiny clusters. No matter if it is near or distant noise, it is given much less weight than a

true data cluster, and does not influence the value of the WMD much, neither influence the decision hyperplane. Therefore, SLMMs are potentially more robust than SVMs and M^4s.

For the near noise situation in Fig. 11(a), the classification accuracies on testing data (400 points) for the SLMM, the SVM and the M^4 are 100%, 97.75%, and 99.25%, respectively, while for the distant noise situation in Fig. 11(b), the test accuracies are 100%, 100%, and 99.25%, respectively.

5.3 On scalability

SLMMs can be extended to have a stronger scalability. After data structure information is extracted, all samples, except those near the decision boundary, can be simply ignored as they are no longer helpful in the decision hyperplane determination (Wang et al. 2005). In SLMMs, each data point corresponds to a constraint in the optimization problem. Therefore, when the number of constraints is reduced, the complexity of SLMMs is reduced accordingly.

The data reduction strategy can be summarized in three steps.

1. *Find the data structures for the positive class and the negative class, respectively.* For example, given the training dataset as plotted in Fig. 12(a), two clusters in the negative class and one cluster in the positive class are detected, which are represented by circular and triangular points, respectively.
2. *Remove the interior samples in each cluster.* For each cluster, calculate the Mahalanobis distance from each data pattern in it to the cluster itself. Then pick out patterns whose Mahalanobis distances are larger than a threshold, and remove all the other data points, as shown in Fig. 12(b).
3. *For each cluster, remove exterior samples that are distant from the opposite class.* Some of the exterior points of clusters can be further removed if their distances to the clusters in the opposite class are greater than the average distance. For instance, in Fig. 12(c), R_{N_1} and R_{N_2} are average Mahalanobis distances from P_1 to N_1 and N_2. Curves I_1 and I_2 are isolines, on which the Mahalanobis distances to N_1 and N_2 are equal to R_{N_1} and R_{N_2}, respectively. Therefore, data points within these two lines in P_1 are selected as potential support vectors, while other exterior points in P_1 will be eliminated. The same procedure will also be applied to N_1 and N_2, and the final result after deleting all unnecessary samples is shown in Fig. 12(d).

In order to illustrate its potential scalability, we test the SLMM model with the above-introduced data reduction strategy on the IMAGE dataset. By changing the interior data reduction threshold in the second step (cf. Fig. 12(b)), pairs of dataset size and testing accuracy, as well as pairs of dataset size and training time, are achieved, as plotted in Fig. 13. The testing accuracy curve (cf. Fig. 13(a)) remains rather flat until a very large proportion (almost 90%) of training data are removed, while the training time (cf. Fig. 13(b)) drops much faster. That means we can find an appropriate cutting point of the training set size, which dramatically shortens the training time while maintaining the classification quality.

5.4 Is the data distribution information always reliable?

The main motivation for us to propose a structured large margin learning model, SLMM, is the desire to further improve the generalization ability of a family of popular classifiers, i.e., large margin machines, by artfully embedding the detected data distribution information into the determination of the separating hyperplane.

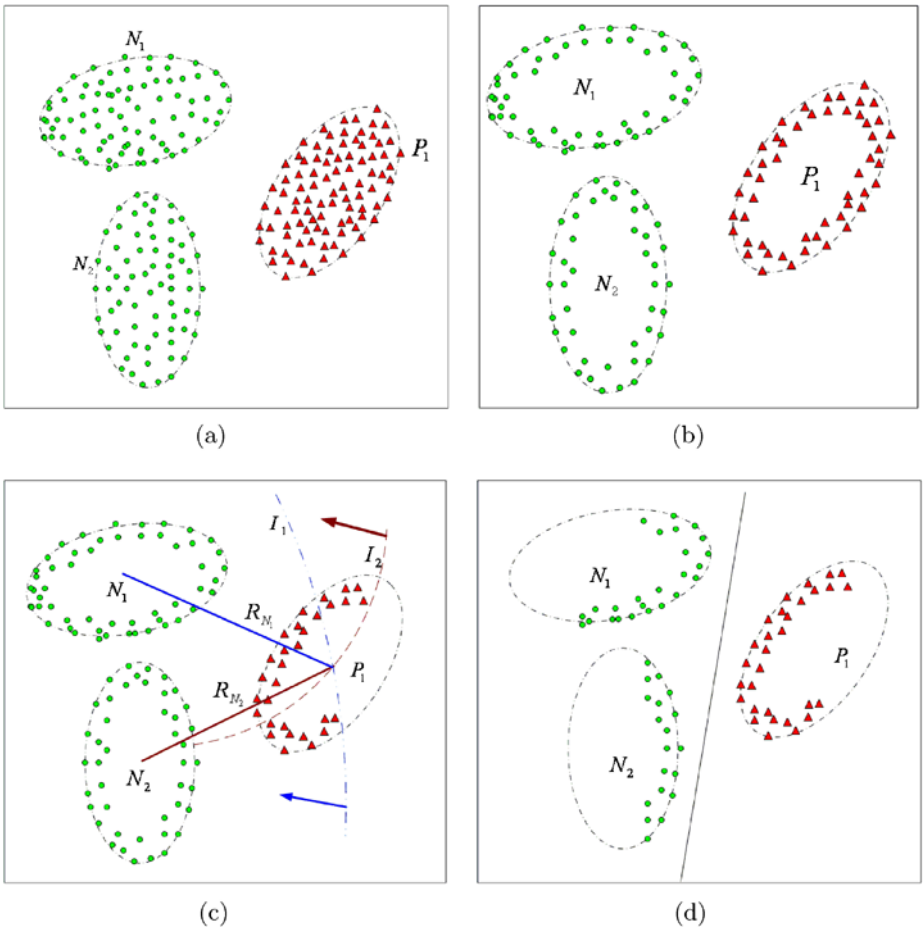
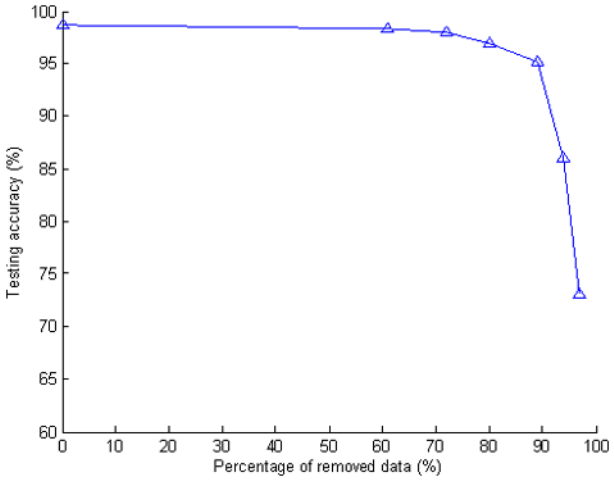
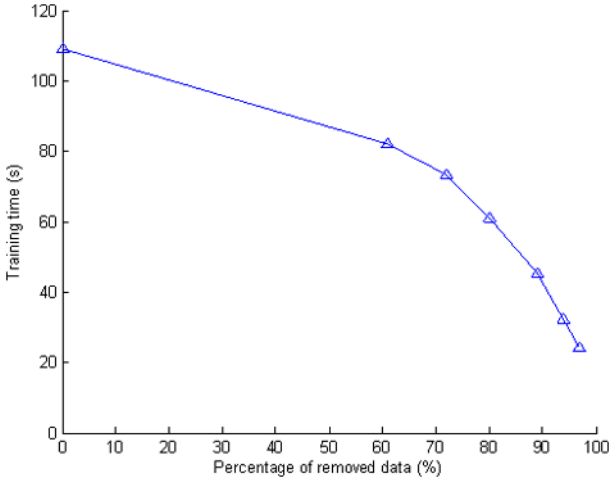


Fig. 12 A geometric illustration of the data reduction process via data structure analysis for SLMMs. **a** After clustering: clusters N_1 and N_2 in the negative class and cluster P_1 in the positive class. **b** After removing the interior data points in each cluster. **c** After removing the exterior data points that are not likely to be support vectors in P_1. **d** After removing the exterior data points that are not likely to be support vectors in P_1, N_1 and N_2, and the resulting decision boundary learned from the reduced training set

It is also worth noticing that a "ground truth" for data distribution may be hard to precisely define, let alone verify. Suppose we find such a distribution, then an optimal classifier can immediately be found. Because of the slight ambiguity and conditionality in the understanding of "real" data distribution, it is not always safe to construct the classifier only based on the "structures" detected from the training data. The RBFN model overemphasizes the role of data clusters, so its generalization ability is limited. The SLMM is such a model in which the data distribution information is neither ignored (because it uses the clustering technique to find out the data structure) nor overemphasized (because of the large margin constraint), but appropriately considered.



(a)



(b)

Fig. 13 Evaluation of the data reduction strategy on the IMAGE dataset. **a** The testing accuracy vs. the percentage of removed data. **b** The training time vs. the percentage of removed data

6 Conclusion and future directions

This paper presents a new large margin classifier, the structured large margin machine, with merits of structured learning. Because it incorporates the data distribution information by using the weighted Mahalanobis distance as its distance metric, SLMMs are more capable than SVMs and M^4s in correctly classifying the unseen patterns. The experimental results demonstrate the high utility of the SLMM model and also prove its robustness. The motivation for SLMMs is triggered by studying existing machine learning approaches from a new angle, which categorizes learning models into structured and unstructured learning. Under the "structured learning" framework, more "structured" classifiers could be constructed by taking the data structure information into account.

As discussed in Sect. 5, the SLMM is potentially a versatile learning model as it possesses extra advantages, e.g., the noise tolerance ability and the strong scalability. Actually, there are still other potential extensions of SLMMs.

- In the semi-supervised learning task, only part of the samples are labeled. As we initially perform data clustering in SLMMs, the unlabeled samples could be assigned with the label of the majority in the cluster it belongs to. Then these newly-labeled samples are used to train the classifier together with the originally labeled samples. In this way, SLMMs can be straightforwardly applied in semi-supervised classification. Computational complexity could be further reduced by integrating the data reduction technique mentioned in Sect. 5.3.
- The SLMM model we discussed in this paper just focuses on binary classification problems, but the multi-class classification tasks can also be solved using the decision directed acylic graph (DDAG) (Hsu and Lin 2002; Platt et al. 2000) combination of pairwise SLMMs.

There are still research opportunities concerning this structured learning model. For example, although we can reduce the training data for SLMMs, the training efficiency is still limited by the complexity of SOCP solving; thus currently this model cannot be applied to learning problems where high efficiency is demanded. Therefore, an efficient and dedicated method for solving the optimization problem in SLMMs is worthy of deep investigation. The derivation of the generalization error bound of SLMMs is also left as future work. Moreover, the potential extensions of SLMMs deserve further research efforts to substantiate and validate.

Acknowledgements The authors would like to thank Dr. K. Huang for the helpful discussions as well as the anonymous reviewers for their valuable and incisive comments, which greatly improved the paper.

Appendix 1 Proof of Theorem 1

Assume $\mathbf{w} = \mathbf{w}_p + \mathbf{w}_q$, where $\mathbf{w}_p$is the projection of $\mathbf{w}$ in the vector space spanned by the kernel maps of all training patterns, and $\mathbf{w}_q$ is the perpendicular component to this vector space. Then the optimization problem in the SLMM model is

$$\max \quad \rho - C \sum_{\ell=1}^{|P|+|N|} \xi_\ell \tag{7a}$$

$$\text{s.t.} \quad ((\mathbf{w}_p + \mathbf{w}_q)^T \Phi(\mathbf{x}_\ell) + b) \geq \frac{|P_i|}{Max_P} \rho \sqrt{(\mathbf{w}_p + \mathbf{w}_q)^T \mathbf{\Sigma}_{P_i}^{\Phi} (\mathbf{w}_p + \mathbf{w}_q)} - \xi_\ell,$$

$$\mathbf{x}_\ell \in P_i, \tag{7b}$$

$$-((\mathbf{w}_p + \mathbf{w}_q)^T \Phi(\mathbf{x}_\ell) + b) \geq \frac{|N_j|}{Max_N} \rho \sqrt{(\mathbf{w}_p + \mathbf{w}_q)^T \mathbf{\Sigma}_{N_j}^{\Phi} (\mathbf{w}_p + \mathbf{w}_q)} - \xi_\ell,$$

$$\mathbf{x}_\ell \in N_j, \tag{7c}$$

$$\xi_\ell \geq 0. \tag{7d}$$

According to this assumption, $\mathbf{w}_q$ is orthogonal to $\Phi(\mathbf{x}_\ell)$ and $\Phi(\mathbf{r})$, where $\Phi(\mathbf{x}_\ell)$ is the image of training pattern $\mathbf{x}_\ell$. Therefore, we have $\mathbf{w}_q^T \Phi(\mathbf{x}_\ell) = \mathbf{w}_q^T \boldsymbol{\mu}_{P_i}^{\Phi} = \mathbf{w}_q^T \boldsymbol{\mu}_{N_j}^{\Phi} = 0$, where $\boldsymbol{\mu}_{P_i}^{\Phi}$ and $\boldsymbol{\mu}_{N_j}^{\Phi}$ are the means of the ith cluster in the positive class and the jth cluster in the

negative class, respectively. By substituting these equations into the optimization problem (7a–7d), we immediately obtain

$$\max \quad \rho - C \sum_{\ell=1}^{|P|+|N|} \xi_\ell \tag{8a}$$

$$\text{s.t.} \quad (\mathbf{w}_p^T \Phi(\mathbf{x}_\ell) + b) \geq \frac{|P_i|}{Max_P} \rho \sqrt{\mathbf{w}_p^T \mathbf{\Sigma}_{P_i}^{\Phi} \mathbf{w}_p} - \xi_\ell, \quad \mathbf{x}_\ell \in P_i, \tag{8b}$$

$$-(\mathbf{w}_p^T \Phi(\mathbf{x}_\ell) + b) \geq \frac{|N_j|}{Max_N} \rho \sqrt{\mathbf{w}_p^T \mathbf{\Sigma}_{N_j}^{\Phi} \mathbf{w}_p} - \xi_\ell, \quad \mathbf{x}_\ell \in N_j, \tag{8c}$$

$$\mathbf{w}_p^T \Phi(\mathbf{r}) = 1, \tag{8d}$$

$$\xi_\ell \geq 0. \tag{8e}$$

Inspecting the difference between the optimization problems (7a–7d) and (8a–8e), one can easily find that $\mathbf{w}_p + \mathbf{w}_q = \mathbf{w}_p$, i.e., $\mathbf{w}_q = 0$. This means that the optimal $\mathbf{w}$ falls in the vector space spanned by the maps of all the training patterns.

Appendix 2 Derivation of the optimization problem in the kernelized SLMM

We first, for convenience of presentation, recall the original optimization problem for the nonlinear SLMM is

$$\max \quad \rho - C \sum_{\ell=1}^{|P|+|N|} \xi_\ell \tag{9a}$$

$$\text{s.t.} \quad (\mathbf{w}^T \Phi(\mathbf{x}_\ell) + b) \geq \frac{|P_i|}{Max_P} \rho \sqrt{\mathbf{w}^T \mathbf{\Sigma}_{P_i}^{\Phi} \mathbf{w}} - \xi_\ell, \quad \mathbf{x}_\ell \in P_i, \tag{9b}$$

$$-(\mathbf{w}^T \Phi(\mathbf{x}_\ell) + b) \geq \frac{|N_j|}{Max_N} \rho \sqrt{\mathbf{w}^T \mathbf{\Sigma}_{N_j}^{\Phi} \mathbf{w}} - \xi_\ell, \quad \mathbf{x}_\ell \in N_j, \tag{9c}$$

$$\xi_\ell \geq 0. \tag{9d}$$

According to Theorem 1, $\mathbf{w}$ can be expressed as

$$\mathbf{w} = \sum_{k=1}^{|P|+|N|} \alpha_k \Phi(\mathbf{x}_k), \quad \alpha_k \in \mathbb{R}. \tag{10}$$

One can easily verify

$$\mathbf{w}^T \Phi(\mathbf{x}_\ell) = \left(\sum_{k=1}^{|P|+|N|} \alpha_k \Phi(\mathbf{x}_k) \right)^T \Phi(\mathbf{x}_\ell) = \sum_{k=1}^{|P|+|N|} \alpha_k K(\mathbf{x}_k, \mathbf{x}_\ell) = \mathbf{K}_\ell \boldsymbol{\alpha}, \tag{11}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{|P|+|N|}]^T$, $\mathbf{K}_\ell$ is the ℓ^{th} row in the kernel Gram matrix $\mathbf{K}$, and $\mathbf{K}(i, j) := K(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, |P| + |N|$.

$\mathbf{A}_{P_i}^{\Phi}$ is a matrix with $|P_i|$ rows, in which the s^{th} row is $\Phi(\mathbf{x}_s)^T$, a kernel map of the s^{th} point $\mathbf{x}_s$ in cluster P_i. With the same size as $\mathbf{A}_{P_i}^{\Phi}$, $\mathbf{M}_{P_i}^{\Phi}$ is the mean matrix of cluster P_i. Each

row in $\mathbf{M}_{P_i}^{\Phi}$ is

$$\boldsymbol{\mu}_{P_i}^{\Phi} = \frac{1}{|P_i|} \sum_{\mathbf{x}_s \in P_i} \Phi(\mathbf{x}_s). \tag{12}$$

Then the covariance matrix for cluster P_i can be written as

$$\boldsymbol{\Sigma}_{P_i}^{\Phi} = \frac{1}{|P_i|} (\mathbf{A}_{P_i}^{\Phi} - \mathbf{M}_{P_i}^{\Phi})^T (\mathbf{A}_{P_i}^{\Phi} - \mathbf{M}_{P_i}^{\Phi}). \tag{13}$$

So we obviously have

$$\mathbf{w}^T \boldsymbol{\Sigma}_{P_i}^{\Phi} \mathbf{w} = \left(\frac{1}{\sqrt{|P_i|}} (\mathbf{A}_{P_i}^{\Phi} - \mathbf{M}_{P_i}^{\Phi}) \mathbf{w} \right)^T \left(\frac{1}{\sqrt{|P_i|}} (\mathbf{A}_{P_i}^{\Phi} - \mathbf{M}_{P_i}^{\Phi}) \mathbf{w} \right). \tag{14}$$

Considering (10) and (12), we obtain,

$$\begin{aligned} &\frac{1}{\sqrt{|P_i|}} (\mathbf{A}_{P_i}^{\Phi} - \mathbf{M}_{P_i}^{\Phi}) \mathbf{w} \\ &= \frac{1}{\sqrt{|P_i|}} \left(\mathbf{A}_{P_i}^{\Phi} \sum_{k=1}^{|P|+|N|} \alpha_k \Phi(\mathbf{x}_k) - \frac{\sum_{\mathbf{x}_s \in P_i} \Phi(\mathbf{x}_s)}{|P_i|} \sum_{k=1}^{|P|+|N|} \alpha_k \Phi(\mathbf{x}_k) \right) \\ &= \frac{1}{\sqrt{|P_i|}} (\mathbf{K}_{P_i} \boldsymbol{\alpha} + \mathbf{e}_{|P_i|} \mathbf{v}_{P_i} \boldsymbol{\alpha}), \end{aligned} \tag{15}$$

where $\mathbf{K}_{P_i}$ is the kernel Gram matrix between the cluster P_i and all the training patterns, i.e., $\mathbf{K}_{P_i}(s, j) := K(\mathbf{x}_s, \mathbf{x}_j)$, $s = 1, \ldots, |P_i|$, $j = 1, \ldots, |P| + |N|$. $\mathbf{e}_{|P_i|}$ is the all-one column vector with length $|P_i|$. $\mathbf{v}_{P_i}$ is the mean vector of matrix $\mathbf{K}_{P_i}$, i.e., $\mathbf{v}_{P_i}(j) = \frac{1}{|P_i|} \sum_{\mathbf{x}_s \in P_i} K(\mathbf{x}_s, \mathbf{x}_j)$, $j = 1, \ldots, |P| + |N|$.

In fact, we can further simplify (15) as

$$\frac{1}{\sqrt{|P_i|}} (\mathbf{A}_{P_i}^{\Phi} - \mathbf{M}_{P_i}^{\Phi}) \mathbf{w} = \tilde{\mathbf{K}}_{P_i} \boldsymbol{\alpha}, \tag{16}$$

where $\tilde{\mathbf{K}}_{P_i} = \frac{1}{\sqrt{|P_i|}} (\mathbf{K}_{P_i} - \mathbf{e}_{|P_i|} \cdot \mathbf{v}_{P_i}^T)$.

Substituting (16) in (14) leads to

$$\mathbf{w}^T \boldsymbol{\Sigma}_{P_i}^{\Phi} \mathbf{w} = \boldsymbol{\alpha}^T \tilde{\mathbf{K}}_{P_i}^T \tilde{\mathbf{K}}_{P_i} \boldsymbol{\alpha}. \tag{17}$$

Similarly, for the negative cluster N_j, we have

$$\mathbf{w}^T \boldsymbol{\Sigma}_{N_j}^{\Phi} \mathbf{w} = \boldsymbol{\alpha}^T \tilde{\mathbf{K}}_{N_j}^T \tilde{\mathbf{K}}_{N_j} \boldsymbol{\alpha}. \tag{18}$$

By adding the constraint (3d) to limit the magnitude of $\boldsymbol{\alpha}$, and replacing the relevant terms in the optimization problem (9a–9d) with (11), (17), and (18), we have (3a–3e).

Appendix 3 Derivation of the Ward's linkage updating formula in the kernel space

Suppose A^{Φ}, B^{Φ} and C^{Φ} are clusters in the kernel space, where A^{Φ} and B^{Φ} are combined together to form a larger cluster A'^{Φ}, i.e., $A'^{\Phi} = A^{\Phi} \cup B^{\Phi}$. We now derive the formula for the new Ward's linkage between cluster A'^{Φ} and C^{Φ}.

According to the Ward's linkage definition, we have

$$W(A'^{\Phi}, C^{\Phi}) = \frac{|A'^{\Phi}| \cdot |C^{\Phi}|}{|A'^{\Phi}| + |C^{\Phi}|} \|\boldsymbol{\mu}_{A'^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2, \tag{19}$$

where $\boldsymbol{\mu}_{S^{\Phi}}$ represents the mean of cluster S^{Φ} in the kernel space.

Replacing $\boldsymbol{\mu}_{A'^{\Phi}}$ with $\frac{|A^{\Phi}|\boldsymbol{\mu}_{A^{\Phi}} + |B^{\Phi}|\boldsymbol{\mu}_{B^{\Phi}}}{|A^{\Phi}| + |B^{\Phi}|}$ in (19), we immediately have

$$\begin{aligned} W(A'^{\Phi}, C^{\Phi}) &= \frac{(|A^{\Phi}| + |B^{\Phi}|) \cdot |C^{\Phi}|}{|A^{\Phi}| + |B^{\Phi}| + |C^{\Phi}|} \left\| \frac{|A^{\Phi}|\boldsymbol{\mu}_{A^{\Phi}} + |B^{\Phi}|\boldsymbol{\mu}_{B^{\Phi}}}{|A^{\Phi}| + |B^{\Phi}|} - \boldsymbol{\mu}_{C^{\Phi}} \right\|^2 \\ &= \frac{|C^{\Phi}|}{(|A^{\Phi}| + |B^{\Phi}| + |C^{\Phi}|) \cdot (|A^{\Phi}| + |B^{\Phi}|)} \\ &\quad \times \| |A^{\Phi}|(\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}) + |B^{\Phi}|(\boldsymbol{\mu}_{B^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}) \|^2. \end{aligned} \tag{20}$$

In fact,

$$\begin{aligned} &\| |A^{\Phi}|(\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}) + |B^{\Phi}|(\boldsymbol{\mu}_{B^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}) \|^2 \\ &= |A^{\Phi}|^2 \|\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2 + |B^{\Phi}|^2 \|\boldsymbol{\mu}_{B^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2 \\ &\quad + |A^{\Phi}||B^{\Phi}|(\|\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2 + \|\boldsymbol{\mu}_{B^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2 - \|\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{B^{\Phi}}\|^2) \\ &= |A^{\Phi}|^2 \frac{|A^{\Phi}| + |C^{\Phi}|}{|A^{\Phi}||C^{\Phi}|} W(A^{\Phi}, C^{\Phi}) + |B^{\Phi}|^2 \frac{|B^{\Phi}| + |C^{\Phi}|}{|B^{\Phi}||C^{\Phi}|} W(B^{\Phi}, C^{\Phi}) \\ &\quad + |A^{\Phi}||B^{\Phi}| \left(\frac{|A^{\Phi}| + |C^{\Phi}|}{|A^{\Phi}||C^{\Phi}|} W(A^{\Phi}, C^{\Phi}) + \frac{|B^{\Phi}| + |C^{\Phi}|}{|B^{\Phi}||C^{\Phi}|} W(B^{\Phi}, C^{\Phi}) \right. \\ &\quad \left. - \frac{|A^{\Phi}| + |B^{\Phi}|}{|A^{\Phi}||B^{\Phi}|} W(A^{\Phi}, B^{\Phi}) \right), \end{aligned} \tag{21}$$

where $W(A^{\Phi}, C^{\Phi}) = \frac{|A^{\Phi}| \cdot |C^{\Phi}|}{|A^{\Phi}| + |C^{\Phi}|} \|\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2$, $W(B^{\Phi}, C^{\Phi}) = \frac{|B^{\Phi}| \cdot |C^{\Phi}|}{|B^{\Phi}| + |C^{\Phi}|} \|\boldsymbol{\mu}_{B^{\Phi}} - \boldsymbol{\mu}_{C^{\Phi}}\|^2$, and $W(A^{\Phi}, B^{\Phi}) = \frac{|A^{\Phi}| \cdot |B^{\Phi}|}{|A^{\Phi}| + |B^{\Phi}|} \|\boldsymbol{\mu}_{A^{\Phi}} - \boldsymbol{\mu}_{B^{\Phi}}\|^2$.

Substituting (21) into (20), one can easily obtain

$$\begin{aligned} &W(A'^{\Phi}, C^{\Phi}) \\ &= \frac{(|A^{\Phi}| + |C^{\Phi}|)W(A^{\Phi}, C^{\Phi}) + (|B^{\Phi}| + |C^{\Phi}|)W(B^{\Phi}, C^{\Phi}) - |C^{\Phi}|W(A^{\Phi}, B^{\Phi})}{|A^{\Phi}| + |B^{\Phi}| + |C^{\Phi}|}. \end{aligned}$$

Appendix 4 Proof of no solution existing for the optimization problem (5a–5c)

Suppose ρ^* is the maximal solution to the optimization problem (5a–5c). $\mathbf{w}^*$ and b^* are the corresponding weight and bias. We increase ρ^* by a times ($a > 1$) i.e., $\rho' = a\rho^*$. For ρ', we have

$$\begin{cases} \mathbf{w}' = \dfrac{a\mathbf{w}^*}{\sqrt{(\mathbf{w}^*)^T \boldsymbol{\Sigma}_P \mathbf{w}^*}}, \\ b' = \dfrac{ab^*}{\sqrt{(\mathbf{w}^*)^T \boldsymbol{\Sigma}_P \mathbf{w}^*}}. \end{cases}$$

The pair of $\mathbf{w}'$ and b' still satisfies the constrains (5b) and (5c), i.e.,

$$\begin{cases} \dfrac{(\mathbf{w}'^T\mathbf{x}_\ell + b')}{\sqrt{\mathbf{w}'^T\boldsymbol{\Sigma}_P\mathbf{w}'}} \geq \rho, & \mathbf{x}_\ell \in P, \\ \dfrac{-(\mathbf{w}'^T\mathbf{x}_\ell + b')}{\sqrt{\mathbf{w}'^T\boldsymbol{\Sigma}_N\mathbf{w}'}} \geq \rho, & \mathbf{x}_\ell \in N. \end{cases}$$

The fact $\rho > \rho^*$ contradicts with the assumption that ρ^* is the maximal solution to the optimization problem, so there is no solution for the optimization problem (5a–5c).

References

Andersen, E. D., & Andersen, A. D. (2001). The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In *High performance optimization* (pp. 197–232). Dordrecht: Kluwer Academic.

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

Christodoulou, C., & Pattichis, C. (1999). Unsupervised pattern recognition for the classification of EMG signals. *IEEE Transactions on Biomedical Engineering*, *46*(2), 169–178.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Berlin: Springer.

Duda, R., Hart, P., & Stork, D. G. (2001). *Pattern classification* (2nd edn.). New York: Wiley.

El-Hamdouchi, A., & Willett, P. (1989). Comparison of hierarchic agglomerative clustering methods of document retrieval. *The Computer Journal*, *32*(3), 220–227.

Everitt, S., Landau, S., & Leese, M. (2001). *Cluster analysis*. London: Hodder.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, *13*(3), 780–784.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*. New Jersey: Prentice Hall.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, *13*(2), 415–425.

Huang, K., Yang, H., King, I., & Lyu, M. R. (2004a). Learning large margin classifiers locally and globally. In: *Proceedings of the 21st international conference on machine learning (ICML)'04*, Banff, Canada.

Huang, K., Yang, H., King, I., Lyu, M. R., & Chan, L. (2004b). Minimum error minimax probability machine. *Journal of Machine Learning Research*, *5*, 1253–1286.

Jain, A. K., & Dubes, R. (1988). *Algorithms for clustering data*. New Jersey: Prentice Hall.

Lanckriet, G. R. G., Ghaoui, L. E., Bhattacharyya, C., & Jordan, M. I. (2002). A robust minimax approach to classification. *Journal of Machine Learning Research*, *3*, 555–582.

Lobo, M., Vandenberghe, L., Boyd, S., & Lebret, H. (1998). Applications of second-order cone programming. *Linear Algebra and its Applications*, *284*, 193–228.

Mukherjee, B., Heberlein, L., & Levitt, K. (1994). Network intrusion detection. *IEEE Transactions on Neural Networks*, *8*(3), 26–41.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: an application to face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)'97* (pp. 130–136).

Platt, J., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAGS for multiclass classification. *Advances in Neural information Processing Systems*, *12*, 547–553.

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

Salvador, S., & Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE international conference on tools with AI* (pp. 576–584).

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G., & Smola, A. (1999). Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, *10*(5), 1000–1017.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Smola, A., Bartlett, P., Schölkopf, B., & Schuurmans, D. (2000). *Advances in large margin classifiers*. Cambridge: MIT.

Sturm, J. (1999). Using Sedumi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, *11*, 625–653.

Vapnik, V. N. (1999). *The nature of statistical learning theory*. New York: Springer.

Veeramachaneni, S., & Nagy, G. (2005). Style context with second-order statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(1), 14–22.

Wang, D., Yeung, D. S., & Tsang, E. (2005). Sample reduction for SVMs via data structure analysis. In *Proceedings of the IEEE international conference on system, man, and cybernetics SMC'05* (pp. 1030–1035), Hawaii, USA.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244.