www.s2is.org

International Journal on Smart Sensing and Intelligent Systems

ISSN 1178-5608

NEW INTELLIGENT CLASSIFICATION METHOD BASED ON IMPROVED MEB ALGORITHM

Yongqing Wang, Lei Liu

Department of Computer Science and Applications,

Zhengzhou Institute of Aeronautical Industry Management,

Zhengzhou 450015, China

Email: wyq-yongqing@163.com

Submitted: Oct. 14, 2013 ***Accepted: Feb. 7, 2014*** ***Published: Mar. 1, 2014***

Abstract- We present a simple approximate algorithm to compute the Minimum Enclosing Ball (MEB) of training samples in high dimensional Euclidean space. We prove theoretically that the proposed algorithm converges to the optimum within any precision quickly. Compared to popular MEB algorithms, it has the competitive performances on both training time and accuracy. Besides, the proposed algorithm does not need any extra requirement on kernels, it can be linked with extensive kernel methods, consequently. We also use the proposed algorithm to handle Binary Classification, Multi-class Classification, and Image Clustering problems. Experiments on both synthetic and real-world data sets demonstrate the validity of the algorithm we proposed.

Index terms***:*** **Minimum enclosing ball, approximate algorithm, kernel methods, classification, image clustering.**

I. INTRODUCTION

With the popularization and development of computer science, machine learning has become an important branch of artificial intelligence (AI). Being one important research content and method of AI, classification problem has made many scientific research fruits, and achieved lots of successful applications in more and more fields, such as machine vision, image recognition, information retrieval, text classification, speech recognition, and so on.

a. Review on classification

So far, the methods of classification can be roughly divided into the following three classes.

(1) The first is classical statistical forecasting methods. Statistics is one of the important theoretical foundations of existing machine learning methods. In this method, the related form of parameters in the model is known, using the training samples to estimate parameters needs the form of samples' distribution to be known, which has great limitations. In addition, the traditional statistical research is the gradual theory when sample size tends to infinity, but in the actual problems, the sample size is limited, so some excellent theoretically statistical learning methods do not perform well in actual applications.

(2) The second is experiential nonlinear methods, such as the artificial neural network. For approximating real values, discrete values, or the vector-valued goal function, this method provides a solution with strong robustness. For certain types of problems, such as learning to explain the complex real-world sensor data, artificial neural network is so far known as the most effective learning way. It establishes the nonlinear model based on known samples, overcoming the difficulties of traditional parameter estimation method. But it is lack of unified mathematical theory, excessive training data fitting and poor generalization performance is also an important problem in the learning of artificial neural network.

(3) The third way is Statistical Learning Theory (SLT), which is a specialized theory for the research of machine learning law under small sample, compared with the traditional statistics. This theory developed a new theoretical system on statistical problems with small samples, under which theory not only the requirements for asymptotic performance, but also the pursuit of the optimal results under the conditions of existing limited information are considered to obtain the statistical inference rules. Vapnik et al. started related researches from the 1960s. By the mid of

1990s, with the development and mature of its theory, meanwhile due to the lack of real progress in theory of the neural network learning method, SLT begins to get more and more extensive attention.

It should be noted that, during the period of 1992 to 1995, based on SLT theory, Vapnik et al. proposed successfully the Support Vector Machine (SVM) method [1, 2, 3]. Compared with the traditional statistical learning methods, SVM method has more solid mathematics theory foundation, which can effectively dealing with high-dimensional data under the condition of limited samples, and has the merits of strong generalization ability, convergence to the global optimal, non-sensitive to dimension, etc. Based on these merits, SVM has become one of the most popular research direction in the field of machine learning, and achieved widely research and application successfully in many fields, such as pattern classification, regression analysis, and estimation of density function, and so on.

b. Current researches on SVM

For the SVM classifier, the aim of training is to find out which samples are support vectors, thus determine the decision function to predict the new samples. As a result, the number of support vector is the main factor affecting the training speed. The kernel matrix of training samples, to be computed and stored by SVM, increases with the square of training samples' number, which becomes the bottleneck of SVM for large-scale problems and limits its application. In recent years, many domestic and foreign scholars are looking for more rapid and efficient algorithms of SVM, which can be used to solve the problem of large sample classification. The existing methods can be roughly divided into the following categories.

b.i Decomposition-based algorithms

These algorithms have the common characteristics described as follows. Dividing the original massive quadratic programming problem into many small sub-problems, according to a certain iterative strategy, solving the sub-problems repeatedly to achieve the approximate solution, by which to gradually converge to the optimal solution of the original problem. For example, the Chunking Algorithm proposed by Boser, Guyon and Vapnik, the Decomposition Algorithm proposed by Osuna et al., the subsequent SVM^{Light} Algorithm, SMO Algorithm by Platt, and LiBSVM by Chang, all of which achieved good effect in application. At present, there are still some scholars working on these decomposition-based strategies to design efficient SVM algorithms.

b.ii Online learning-based algorithms

At present, there are some research results on using SVM to handle online learning problems, where data or model needs to be updated continually. The main strategy in these algorithms can be described as follows. Because of the addition of non-support vector has no effect on decision function, so for the new samples, according to the complementary slack conditions in the optimal theory, to determine whether or not to update Lagrange multipliers in the existing model. For considering all the historical data, these methods have disadvantages of unable to control the number of support vector. Only considering the sparseness of solution of the SVM, can we possibly to reduce the amount of calculation, and shorten the calculation time.

b.iii Deformation-based algorithms

In order to improve the learning speed of standard SVM, except for seeking quick solving quadratic programming methods, there are still many algorithms for the simplified and deformed model. For example, the linear programming SVM, the quadratic loss function SVM, the least squares SVM, the proximal SVM, and so on. In order to improve the generalization ability, there are many methods based on the variant of standard SVM model. For example, fuzzy SVM, multiple kernel SVM, and prior knowledge-incorporated SVM, and so on. These algorithms made beneficial attempts from several aspects to improve the operation performance of SVM, and have achieved good effect in application. But this method has not yet formed a mature theoretical system, there are still many aspects need to be further discussed.

b.iv Parallel-based algorithm

In recent years, there are lots of works devoted to parallel implementation method of training SVM. For example, Collobert [4] proposed parallel hybrid method of SVM, Dong [5] used block diagonal matrix to approximate the original kernel matrix, Zanghirati [6] proposed SVM^{light}

parallel method, Huang [7] proposed modular network realization method of SVM, Cao [8] proposed SMO parallel method, and so on. These methods have achieved obvious improvement for massive data problems in practical application. However, can the local optimal solution of sub-problem guarantee the global optimal solution in parallel algorithm, is still a research topic to be urgently solved.

b.v Data reduction-based algorithm

The lower bound of support vector's number is linear with the number of training sample. Taking some kind of strategy, therefore, by choosing the training samples most likely to be support vector, or deleting the training samples most unlikely to be support vector, or taking the above two methods at the same time to preprocess the training set, can reduce the size of training set, and then accelerates the training process without much loss of precision. But the experimental results show that, if the proportion of the support vector in the training set is large, the generalization ability of data reduction-based algorithm is lower than the standard SVM. Asharaf [9] claimed that, even though the decomposition or data-sampling techniques [10, 11, 12, 13] can help to reduce the complexity of the optimization problem, they are still expensive for use in applications involving large data sets.

The quadratic matrix involved in the above five types of algebraic algorithms are required to be sparse, which results in the need for large memory and training process for many practical problems, where the conditions are not met. And that the series of intuitive interpretation-based geometric algorithms, can relieve the contradiction mentioned above.

c. Current researches on SVM geometry methods

Different from the algebraic algorithm of SVM, which solves the dual problem in transformed feature space, the geometry algorithm of SVM solves the original problem in the sample space. After Bennett and Crisp putting forward the idea of finding a pair of nearest points between the convex hulls of two points' sets, there are many excellent SVM geometry algorithms based on the idea of nearest points. For example, projection method-based Swap algorithm is suitable only for linear separable problem, the rapid geometry iteration algorithm proposed by Keerthi can indirectly solve the inseparable problem, and the reduced convex hull algorithm proposed by

Mavroforakis can directly solve the inseparable problem. In addition, Tsang proved the equivalence between the MEB and SVM, introducing an excellent approximation MEB algorithm to solve the SVM classification and regression problems for massive data, called the Core Vector Machine (CVM) [14], whose time complexity is linear with the sample size, and the space complexity is independent of the number of samples. Therefore, CVM is suitable for handling large data classification problems. Although conceptually simple, a sophisticated numerical solver is required for its implementation, which is computationally expensive when applying on large-scale problems with very large core-sets. After that, the Simpler Core Vector Machines (SCVM) [15] replace the numerical solver with an iterative algorithm, which results in a faster training than CVM with comparable accuracy on massive data sets. However, the Enclosing Ball (EB) problem solved in SCVM requires the ball's radius to be fixed, which limits the SCVM to be only feasible for certain kernels that satisfy $k(\varphi(x_i),\varphi(x_i))=k$, a constant.

In this paper we develop a (1+ε)-approximate algorithm for computing the MEB of a given points set without requirement of any numerical solver. Compared to CVM and SCVM algorithms, it has the competitive performances in both training time and accuracy. Besides, the proposed algorithm does not need any extra requirement on kernels, which guarantees the potential applications in extensive kernel methods, consequently. We also use the proposed SMEB algorithm to handle Binary Classification, Multi-class Classification, and Image Clustering problems. Experiments on both synthetic and real-world data sets demonstrate the validity of the algorithm we proposed.

The rest of this paper is organized as follows. Section II provides a review on MEB problem. In Section III, we give the connection between SVM and MEB. Section IV presents the SMEB algorithm in detail and experimental results. In Section V we conclude this paper.

II. REVIEW ON MEB ALGORITHMS

a. Formulation of MEB

The Minimum Enclosing Ball (MEB) problem can be dated back to 1857, which was firstly proposed by Sylvester to investigate the smallest radius disk enclosing n points on the plane. It has found applications in diverse areas, such as computer graphics, machine learning, facility location and shape fitting problems.

Given a set of data points $S=\{x_i \mid i=1,\ldots,m\}$, where $x_i \in R^d$, the minimum enclosing ball of S (denoted as MEB(S)) is defined as the smallest ball $B(c,R)$ that contains all the points in S, i.e., $B(c,R)=\{x_i \in R^d \mid \|x_i - c\| \le R\}$.

Let k be a kernel function with the associated feature map ϕ, i.e. $k(x_i,x_j)=\langle \phi(x_i),\phi(x_j)\rangle$, where $\langle .,.\rangle$ denotes the inner product. Then the primal MEB problem in the kernel-induced feature space to find the MEB(S) with center c and radius R can be formulated as

$$\begin{aligned}&\min_{R,c} \; R^2 \\ &s.t. \; ||c-\phi(x_i)||^2 \le R^2, i=1,\ldots,m.\end{aligned} \tag{1}$$

The corresponding dual is

$$\begin{aligned}&\max_{\alpha_i} \; \sum_{i=1}^{m}\alpha_i k(x_i,x_i)-\sum_{i,j=1}^{m}\alpha_i\alpha_j k(x_i,x_j) \\ &s.t. \; \sum_{i=1}^{m}\alpha_i = 1, \; \alpha_i \ge 0, \; i=1,\ldots,m.\end{aligned} \tag{2}$$

b. Exact algorithms of MEB

Traditional exact MEB algorithms can not effectively deal with high-dimensional data. For example, prune-and-search algorithm proposed by Megiddo, heuristic-based move-to-front algorithm proposed by Welzl, quadratic programming method proposed by GÄartner, the F-G-K algorithm proposed by Fischer [16]. We briefly present the iteration strategy of F-G-K algorithm below, called the dropping and walking (see Figure 1).

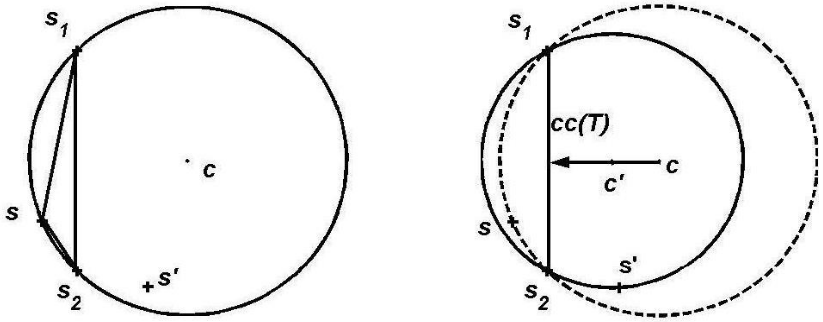


Figure 1. Dropping and walking: Dropping s from $T=\{s,s_1,s_2\}$ (left) and walking towards the center cc(T) of the circumsphere of $T=\{s_1,s_2\}$ until s' stop walking (right).

c. Approximate algorithms of MEB

In recent years, MEB are required to solve the application problem in high dimension space. Therefore, from the perspective of practical application, many scholars are studying how to develop more rapid approximate MEB algorithm. Among them, Badoiu firstly proposed the concept of core set (see Figure 2) to solve the clustering problem in high dimensional. Thereafter, the idea of core set was introduced to improve the performance of MEB in B-K algorithm. In the following, we illustrate the strategy of B-K algorithm in Figure 3 [17].

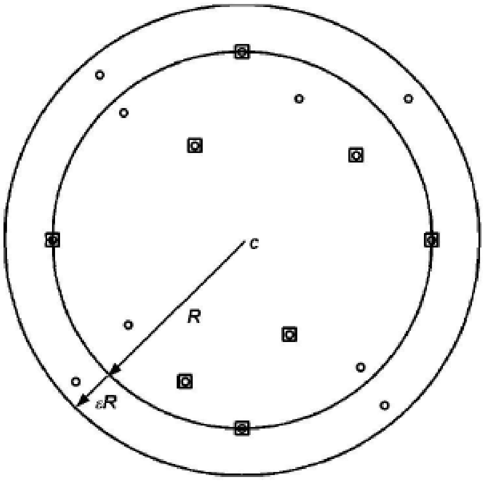


Figure 2. Inner circle is the exact MEB contains all the square points, whose ($1+\varepsilon$) expansion contains all the points, so the square points is the core set of all points.

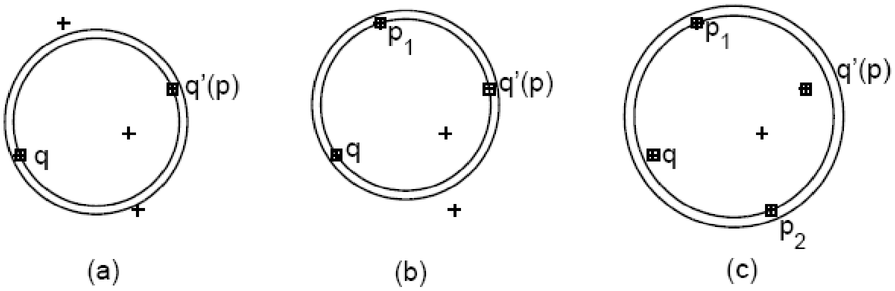


Figure 3. A 3-step of the B-K algorithm achieves a final core-set determined by four points on a five-point set.

III. CONNECTIONS BETWEEN MEB AND SVM

SVM and MEB were firstly connected in Support Vector Data Description (SVDD) in 1999 by Tax and Duin [18], thereafter many related research had been achieved.

a. Support vector data description

We illustrate the idea of SVDD in Figure 4 below. The left figure presents the decisive curve of banana-shaped data without outliers under the parameter C = 1 in two-dimensional plane. The solid points represent the support vectors, the dotted line represents the boundary of the data description, and the grey value represents the distance from the center of sphere, where the deeper means the closer to the center. We can see that, only three support vectors are needed to describe all of the data set. Introducing a new outlier in the right picture (located with an arrow), large change occurred in the decisive curve, where the new outlier comes into the support vector of the new decisive curve.

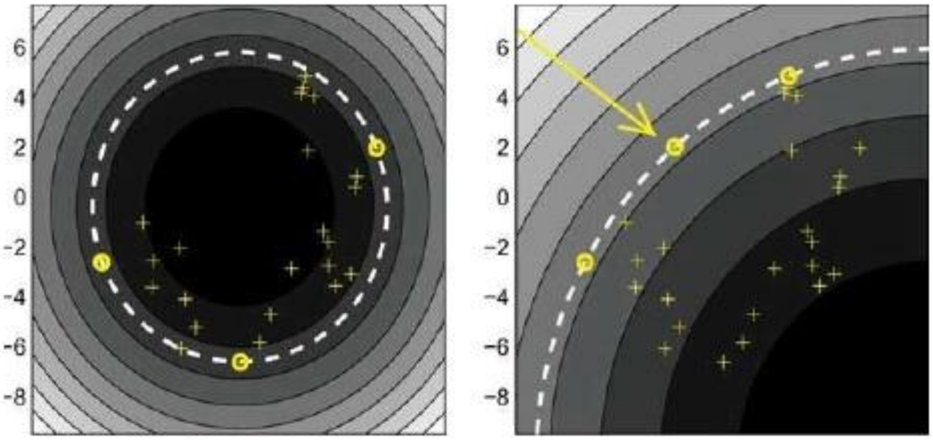


Figure 4. SVDD without outliers (left) and SVDD with outliers (right)

b. Equivalent formulation under normalized kernels

Considering only the situation where the kernel k satisfies $k(x,x)=\kappa$, a constant. This holds true for kernels like Gaussian, polynomial kernel with normalized inputs, and any normalized kernels [14]. Then the dual of the MEB problem (2) can be rewritten as

$$\begin{aligned} \min_{\alpha_i} \quad & \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \\ s.t. \quad & \sum_{i=1}^{m} \alpha_i = 1,\ \alpha_i \geq 0,\ i = 1,\dots,m. \end{aligned} \qquad (3)$$

When the involved kernels fulfill the requirements mentioned above, any Quadratic Programming (QP) of form (3) can be identified as an MEB problem.

b.i Formulating Binary SVM as MEB under Normalized Kernels

Binary SVM can be formulated as a QP to maximize the margin between two classes, and the consequent generalization ability is always better than the other machine learning methods.

Given a training data sets $S = \{(x_i, y_i) \mid i = 1,\dots,m\}$, where $x_i \in R^d$ and $y_i \in \{+1,\text{-}1\}$, the primal for the Binary SVM problem can be formulated as

$$\begin{aligned} &\min_{w,\rho,b,\xi_i} \; ||w||^2 + b^2 - 2\rho + C\sum_{i=1}^{m}\xi_i^{\,2} \\ &s.t. \quad y_i(w'\phi(x_i) + b) \ge \rho - \xi_i, i = 1,\dots,m. \end{aligned} \qquad (4)$$

The corresponding dual is

$$\begin{aligned} &\min_{\alpha_i} \quad \sum_{i,j=1}^{m}\alpha_i\alpha_j(y_i y_j k(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C}) \\ &s.t. \quad \sum_{i=1}^{m}\alpha_i = 1, \;\; \alpha_i \ge 0, \;\; i = 1,\dots,m, \end{aligned} \qquad (5)$$

Where δ_{ij} is the Kronecker delta function, defined as

$$\delta_{ij} = \begin{cases} 1, & if \; i = j, \\ 0, & if \; i \ne j. \end{cases} \qquad (6)$$

We denote the pair (x_i, y_i) as z_i to simplify the notation. Introducing a modified feature map $\tilde{\phi}(z_i) = [y_i\phi'(x_i) \;\; y_i \;\; \frac{e'_i}{\sqrt{C}}]'$ and the associated kernel function $\tilde{k}(z_i, z_j) = y_i y_j k(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C}$, then the dual of Binary SVM with form (5) can be rewritten as

$$\begin{aligned} &\min_{\alpha_i} \quad \sum_{i,j=1}^{m}\alpha_i\alpha_j\tilde{k}(z_i, z_j) \\ &s.t. \quad \sum_{i=1}^{m}\alpha_i = 1, \; \alpha_i \ge 0, \; i = 1,\dots,m. \end{aligned} \qquad (7)$$

Hence the Binary SVM can be viewing as an MEB problem [14].

b.ii Formulating Multi-class SVM as MEB under Normalized Kernels

Traditionally, multi-class pattern recognition problems are typically solved using voting scheme methods based on combining many binary classification decision functions. With the idea of a reinterpretation of the normal vector of the separating hyper-plane, Szedmak and Shawe-Taylor formulated multi-SVM as an SVM with vector output, where the vector can be seen as a projection operator of the feature vectors into a one-dimensional subspace.

Given a set of data points $S=\{(x_i, y_i)\,|\, i=1,\ldots,m\}$, where $x_i \in R^d$, $y_i \in R^T$, i.e., we have m training data whose labels are vector valued. Obviously there are many choices of the vector labels, the simplest is the indicator vectors of the classes following the rule

$$(y_i)_t=\begin{cases}1, & \text{item i belongs to category t, t = 1,...,T,}\\ 0, & \text{otherwise.}\end{cases} \qquad (8)$$

Then the primal of the multi-SVM can be formulated as

$$\begin{aligned}&\min_{w,\rho,b,\xi_i}\ trace(w'w)+\|b\|^2-2\rho+C\sum_{i=1}^{m}\xi_i^{\,2}\\ &s.t.\quad y_i'(w\phi(x_i)+b)\ge\rho-\xi_i,\ i=1,\ldots,m.\end{aligned} \qquad (9)$$

The corresponding dual is

$$\begin{aligned}&\min_{\alpha_i}\ \sum_{i,j=1}^{m}\alpha_i\alpha_j(y_i'y_jk(x_i,x_j)+y_i'y_j+\frac{\delta_{ij}}{C})\\ &s.t.\ \sum_{i=1}^{m}\alpha_i=1,\ \alpha_i\ge 0,\ \ i=1,\ldots,m.\end{aligned} \qquad (10)$$

From the KKT conditions on form (9) we can get $w=\sum_{i=1}^{m}\alpha_i y_i\phi(x_i)'$, $b=\sum_{i=1}^{m}\alpha_i y_i$. So the decision function predicting one of the labels from $\{(y_i)\,|\, i=1,\ldots,m\}$ for any test pattern xj can be descried as

$$\arg\max_{t=1,\ldots,T}\ y_t'(w\phi(x_j)+b)=\arg\max_{t=1,\ldots,T}\ (\sum_{i=1}^{m}\alpha_i y_i' y_t(k(x_i,x_j)+1)). \qquad (11)$$

We denote the pair (x_i, y_i) as z_i to simplify the notation. Introducing a modified feature map $\tilde{\phi}(z_i)=[y_i'\phi'(x_i)\ \ y_i'\ \ \frac{e'_i}{\sqrt{C}}]'$ and associated kernel function $\tilde{k}(z_i,z_j)=y_i'y_jk(x_i,x_j)+y_i'y_j+\frac{\delta_{ij}}{C}$, then the dual of multi-SVM with form (10) can be rewritten as

$$\begin{aligned}&\min_{\alpha_i}\ \sum_{i,j=1}^{m}\alpha_i\alpha_j\tilde{k}(z_i,z_j)\\ &s.t.\ \sum_{i=1}^{m}\alpha_i=1,\ \alpha_i\ge 0,\ \ i=1,\ldots,m.\end{aligned} \qquad (12)$$

Hence the multi-SVM can be viewed as an MEB problem [9].

c. Equivalent formulation under non-normalized kernels

Taking into account that many real-world data sets in binary SVM are imbalanced and the majority class has much more training patterns than the minority class. As a result, the resultant hyper-plane will be shifted towards the majority class. A common remedy is to use different $C's$

for the two classes. In the more general case, each training pattern can have its own penalty factor C_i. The primal is then formulated as

$$\begin{aligned} \min_{w,\rho,b,\xi_i} \ & \|w\|^2 + ||b||^2 - 2\rho + \sum_{i=1}^{m} C_i \xi_i^2 \\ s.t. \quad & y_i(w'\phi(x_i)+b) \geq \rho - \xi_i, i = 1,\ldots,m. \end{aligned} \qquad (13)$$

And the corresponding dual is

$$\begin{aligned} \max_{\alpha_i} \quad & -\alpha'\tilde{K}\alpha \\ s.t. \quad & \alpha'1 = 1, \alpha \geq 0. \end{aligned} \qquad (14)$$

Where $\tilde{k} = [y_i y_j k(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C_i}]$. Note that as the C_i's are different in general, the diagonal entry $\tilde{K}_{ii} = k(x_i, x_i) + 1 + \frac{1}{C_i}$ is not constant even if $k(x_i, x_i)$ is.

Besides, the diagonal entry of kernel matrix $k(x_i, x_i)$ in Ranking SVM is not a constant either, does not satisfy the requirement that the involved kernels need to be normalized, under which cases, CCMEB [19] works.

Instead of finding the smallest ball enclosing all $\varphi(x_i) \in S$ in the feature space described in (1), CCMEB [19] augments an extra $\delta_i \in R$ to each $\varphi(x_i)$, forming $\begin{bmatrix}\varphi(x_i)\\ \delta_i\end{bmatrix}$. Then the task is to find the MEB for these augmented points, meanwhile constraining the last coordinate of the ball's center to be zero (i.e., of the form $\begin{bmatrix}c\\ 0\end{bmatrix}$) (seen Figure 5).

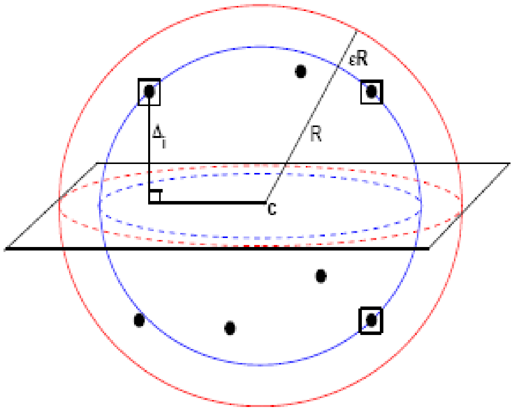


Figure 5. Center-Constrained MEB

The primal in (1) is thus changed to

$$\begin{array}{ll} \min\limits_{R,c} & R^2 \\ s.t. & ||c-\phi(x_i)||^2 + \delta_i^2 \le R^2, i = 1,\ldots,m. \end{array} \qquad (15)$$

It easily shows that the corresponding dual is a QP problem

$$\begin{array}{ll} \max\limits_{\alpha_i} & \alpha'(diag(K)+\Delta) - \alpha' K\alpha \\ s.t. & \alpha'1 = 1, \alpha \ge 0. \end{array} \qquad (16)$$

where $\Delta = [\delta_1^2,\ldots,\delta_m^2]' \ge 0.$ From KKT conditions, we can recovering R and C from the optimal α. And then, for an arbitrary $\eta \in R$, (16) yields the same optimal α as

$$\begin{array}{ll} \max\limits_{\alpha_i} & \alpha'(diag(K)+\Delta-\eta 1) - \alpha' K\alpha \\ s.t. & \alpha'1 = 1, \alpha \ge 0. \end{array} \qquad (17)$$

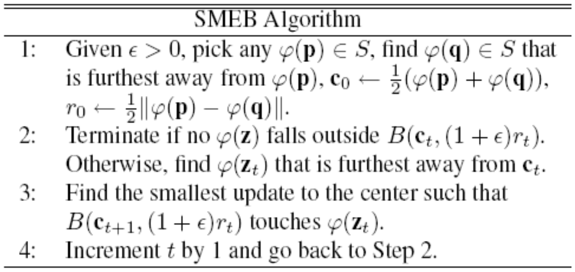
Using the same arguments adopted before, any QP problem of the form (17), with $\Delta \ge 0$, can also be regarded as a MEB problem (15). By defining $\Delta = \max\limits_i\{\tilde{K}_{ii}\}1 - diag(\tilde{K}) \ge 0, \eta = \max\limits_i\{\tilde{K}_{ii}\}$, the linear term in the objective function of (17) disappears. Consequently, all the QPs of (14) and (17) have the same form, without any extra requirement on the kernels. Therefore, the non-normalized kernel methods, such as the binary SVM for imbalanced data and Ranking SVM, can be viewed as a center-constrained MEB problem.

IV. IMPROVED MEB ALGORITHM

In this section we propose the Simpler Minimum Enclosing Ball (SMEB) algorithm to solve classification problems. By dropping the step of setting an appropriate radius of EB in advance in SCVM, we achieve an asymptotically expansive version with a similar process to SCVM without any requirement on the kernels.

a. Detailed procedure of SMEB

Table 1: The procedure of SMEB Algorithm



	SMEB Algorithm
1:	Given $\epsilon > 0$, pick any $\varphi(\mathbf{p}) \in S$, find $\varphi(\mathbf{q}) \in S$ that is furthest away from $\varphi(\mathbf{p})$, $\mathbf{c}_0 \leftarrow \frac{1}{2}(\varphi(\mathbf{p}) + \varphi(\mathbf{q}))$, $r_0 \leftarrow \frac{1}{2}\|\varphi(\mathbf{p}) - \varphi(\mathbf{q})\|$.
2:	Terminate if no $\varphi(\mathbf{z})$ falls outside $B(\mathbf{c}_t, (1+\epsilon)r_t)$. Otherwise, find $\varphi(\mathbf{z}_t)$ that is furthest away from $\mathbf{c}_t$.
3:	Find the smallest update to the center such that $B(\mathbf{c}_{t+1}, (1+\epsilon)r_t)$ touches $\varphi(\mathbf{z}_t)$.
4:	Increment t by 1 and go back to Step 2.

The proposed iterative algorithm is shown in Table 1 above. Unlike the CVM, the update in Step 3 can be performed efficiently without the use of any numerical optimization solver. Unlike the SCVM, the radius in SMEB is asymptotically expansive and converges to the optimum in $O(\frac{1}{\varepsilon})$ iterations within any precision. The efficient update of the t^th iteration is shown in figure 6.

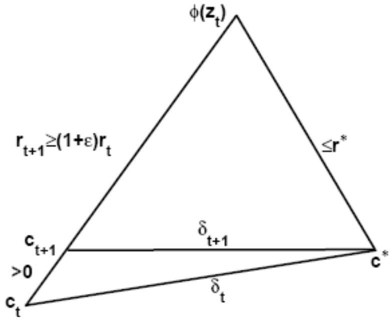


Figure 6. Update of c_t at the t^th iteration

Mathematically, SCVM seeks for the c_{t+1}, r_{t+1} such that

$$\begin{aligned} &\min_{c_{t+1}, r_{t+1}} \quad || c_{t+1} - c_t ||^2 \\ &s.t. \ \ || c_{t+1} - \varphi(z_t) ||^2 \leq r_{t+1}{}^2, \\ &\qquad (1+\varepsilon) r_t \leq r_{t+1}. \end{aligned} \quad (18)$$

From the KKT conditions on form (18) we can get the new center is $c_{t+1} = \beta_t c_t + (1-\beta_t)\varphi(z_t)$, where $\beta_t = \frac{(1+\varepsilon) r_t}{\|c_t - \varphi(z_t)\|} (\geq 0)$, which is a convex combination of c_t and $\varphi(z_t)$. Consequently, for any $t > 0$, c_t is always a convex combination of c_0 and $S_t = \{\varphi(z_i)\}_{i=1}^{t}$, i.e., $c_t = \sum_{i=0}^{t} \alpha_i \varphi(z_i)$, where $\sum_{i=0}^{t} \alpha_i = 1,\ \alpha_i \geq 0,\ \varphi(z_0) = c_0$.

b. Performance analysis of SMEB algorithm

In this Section we conclude that the iterative SMEB algorithm converges to the optimum within any precision, and the time/space complexities are superior to CVM and SCVM.

Proposition 1. SMEB algorithm obtains an $(1+\varepsilon)$-approximation of MEB(S) in $O(\frac{1}{\varepsilon})$ iterations.

Proposition 2. Assume that the SMEB algorithm terminates at the τ^{th} iteration with the solution (r_τ, c_τ), then $\left\| c_\tau - c^* \right\| \leq \sqrt{\varepsilon(2+\varepsilon)} r^*$, where (r^*, c^*) denotes the optimum of MEB(S).

Proposition 3. The time complexity of SMEB algorithm is $O(\frac{m}{\varepsilon^2}+\frac{1}{\varepsilon^3})$, which is linear in m for a fixed ε.

Proposition 4. The space complexity of SMEB algorithm is $O(\frac{1}{\varepsilon^2})$, which is independent of m for a fixed ε.

The detailed proofs of these theorems are omitted here for conciseness, interested readers can refer to Wang [20].

c. Experiments on synthetic data

Experiments are performed on five synthetic data sets, which follow a uniform distribution on the interval (0,10) (Table 2). All experiments do not adopt the probabilistic speedup method utilized in CVM for simplicity. We use Matlab 6.5 on a PC with Pentium-4 3.20 GHz CPU, 1GB of RAM running Windows XP to implement our experiments.

Table 2: Data sets used in the experiments

data sets	**data 1**	**data 2**	**data 3**	**data 4**	**data 5**
dimension	2	2	2	2	2
number	10	100	1000	10000	100000

We compare CVM, SCVM and SMEB on Optimum Bias Ratio and Training Time at different values of ε on the five data sets, where Optimum Bias Ratio (OBR) is defined as $OBR=\frac{\|\text{experimental value - optimum}\|}{\|\text{optimum}\|}$. Table 3 shows that all algorithms have low OBR's for $\varepsilon\in[10^{-6},10^{-4}]$, and SMEB has accuracies comparable with the other two algorithms, especially when ε gets small enough. Table 4 indicates that when utilized to handle larger data set, e.g., data 5, the SMEB algorithm is usually faster than CVM and SCVM for the same value of ε, with comparable accuracies, which implies that SMEB is more suitable for solving larger data problems (Fig. 7). When ε decreases, the SMEB becomes closer to the exact optimal solution, but at the expense of higher time and space complexities. Such a tradeoff between efficiency and approximation quality is typical of all approximation schemes.

Table 3: OBR's of the various MEB algorithms (%)

Algo.	ϵ	data 1	data 2	data 3	data 4	data 5
CVM	10^{-1}	**2.82**	**2.39**	1.88	**0.32**	0.34
	10^{-2}	**0**	**0**	0.21	**0.32**	**0.33**
	10^{-3}	**0**	**0**	**0**	**0**	**0**
	10^{-4}	**0**	**0**	**0**	**0**	**0**
	10^{-5}	**0**	**0**	**0**	**0**	**0**
	10^{-6}	**0**	**0**	**0**	**0**	**0**
SCVM	10^{-1}	7.09	3.95	**1.31**	1.84	2.95
	10^{-2}	4.78	2.6	**3.81e-2**	0.44	0.52
	10^{-3}	0.79	3.39e-2	4.01e-3	0.39	0.21
	10^{-4}	9.04e-2	3.63e-3	3.17e-3	6.31e-2	0.24
	10^{-5}	1.23e-2	**0**	**0**	6.33e-3	1.41e-3
	10^{-6}	3.07e-3	**0**	**0**	**0**	1.41e-3
SMEB	10^{-1}	**2.82**	**2.39**	1.88	**0.32**	**0.33**
	10^{-2}	2.04	0.83	0.21	**0.32**	**0.33**
	10^{-3}	5.97e-2	0.4	9.82e-2	0.12	0.12
	10^{-4}	7.42e-3	6.42e-3	7.53e-2	8.12e-2	9.04e-3
	10^{-5}	3.07e-3	3.63e-3	5.61e-2	6.01e-3	1.41e-3
	10^{-6}	**0**	**0**	1.11e-2	**0**	1.41e-3

Table 4: Training time of the various MEB algorithms (s)

Algo.	ϵ	data 1	data 2	data 3	data 4	data 5
CVM	10^{-1}	0.016	0.032	0.078	2.656	430
	10^{-2}	0.031	0.047	0.110	2.781	447.719
	10^{-3}	0.031	0.047	0.125	2.969	608.375
	10^{-4}	0.047	0.063	0.125	**3.031**	684.891
	10^{-5}	0.047	**0.078**	0.141	**3.421**	775.297
	10^{-6}	**0.047**	**0.078**	**0.157**	**4.594**	1299.609
SCVM	10^{-1}	0.016	**0.015**	**0.031**	**0.844**	408.578
	10^{-2}	0.016	**0.015**	**0.031**	**1.610**	428
	10^{-3}	0.032	**0.016**	0.062	**2.015**	583.813
	10^{-4}	0.14	**0.031**	**0.079**	3.703	589.704
	10^{-5}	0.265	1.906	**0.092**	112.47	598.438
	10^{-6}	0.391	23.207	53.476	2185.36	1207.304
SMEB	10^{-1}	**0.0064**	0.032	**0.031**	1.563	**289.343**
	10^{-2}	**0.0064**	0.046	**0.031**	1.719	**289.719**
	10^{-3}	**0.015**	0.063	**0.047**	8.5	**491.250**
	10^{-4}	**0.016**	0.344	1.734	55.235	**489.718**
	10^{-5}	**0.016**	8.907	17.187	546.266	**498.438**
	10^{-6}	0.156	31.093	173.484	1516.6	**1046.281**

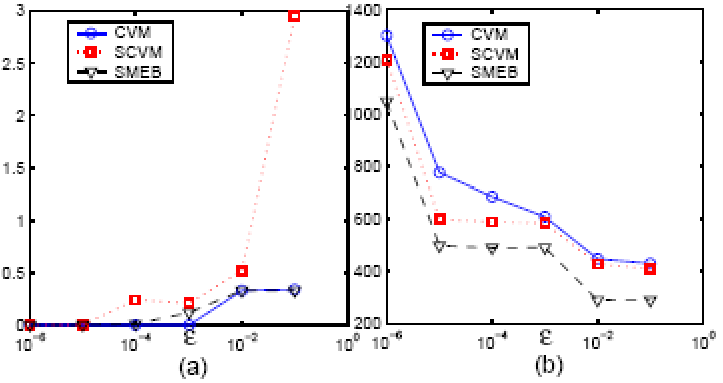
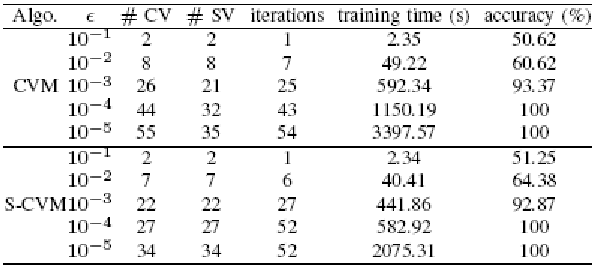


Figure 7. Performance with different value of ε for data 5, (a) for Optimum Bias Ratio (%), (b) for training time (s).

d. Application of SMEB in binary classification

We implemented CVM by utilizing the SMEB algorithm we proposed to handle large scale Binary SVM with smaller core sets (S-CVM). We generated synthetic 4×4 check-board data set, whose points follow the uniform distribution. We compare the performances of CVM and S-CVM on the core sets' size, training time and training accuracy with different value of ε and m on synthetic data sets (Table 5 and Table 6). In all the experiments we used Gaussian kernel function k(u,v)=exp(-p_1(u-v)'(u-v))+ p_2, where the parameters were chosen by grid search as p_1=1, p_2=0, penalty factor C=100.

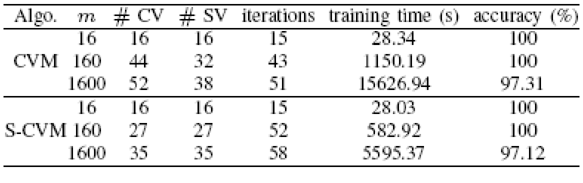
Table 5: Comparison on performances of CVM and S-CVM with different ε in the case of data number m=160



Algo.	ϵ	# CV	# SV	iterations	training time (s)	accuracy (%)
	10^{-1}	2	2	1	2.35	50.62
	10^{-2}	8	8	7	49.22	60.62
CVM	10^{-3}	26	21	25	592.34	93.37
	10^{-4}	44	32	43	1150.19	100
	10^{-5}	55	35	54	3397.57	100
	10^{-1}	2	2	1	2.34	51.25
	10^{-2}	7	7	6	40.41	64.38
S-CVM	10^{-3}	22	22	27	441.86	92.87
	10^{-4}	27	27	52	582.92	100
	10^{-5}	34	34	52	2075.31	100

Observing from Table 5, we find that the size of final core set of S-CVM is smaller than that of CVM at the same ε, and so does the training time. The numbers of support vectors and training accuracies of both algorithms are comparative. Besides, we can see that the training accuracies cannot be enhanced when $\varepsilon \le 10^{-4}$, so the value of ε we adopt in this article is $\varepsilon = 10^{-4}$.

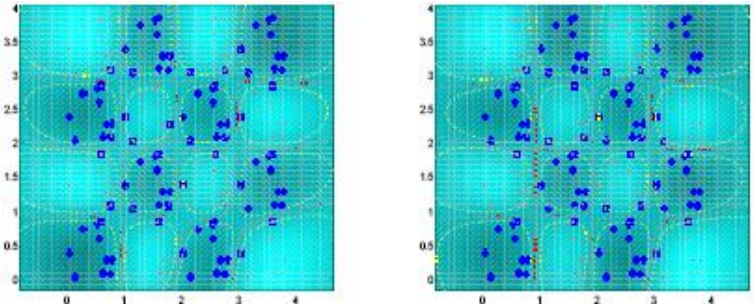
Table 6: Comparison on performances of CVM and S-CVM with different m in the case of $\varepsilon = 10^{-4}$



Algo.	m	# CV	# SV	iterations	training time (s)	accuracy (%)
	16	16	16	15	28.34	100
CVM	160	44	32	43	1150.19	100
	1600	52	38	51	15626.94	97.31
	16	16	16	15	28.03	100
S-CVM	160	27	27	52	582.92	100
	1600	35	35	58	5595.37	97.12

We observing from Table 6 that, the size of final core set of S-CVM is smaller than that of CVM at the same m and so does the training time. The numbers of support vectors and training accuracies of both algorithms are comparative. Besides, with the decrease of ε, the rest of training performances increase, which is a typical feature of approximate algorithm. It is worthy of noticing from Table 6 that, for larger data sets, S-CVM performs more well than CVM. For instance, S-CVM can shorten the training time almost one half for m=160, and two thirds for m=1600 with comparative accuracy.

Figure 8 demonstrates the different performances of CVM and S-CVM under the best value of $\varepsilon = 10^{-4}$ on 4×4 check-board data, where the square points denote the support vectors, yellow dotted lines denote the hyper-plane the support vectors lie in, and the red dotted lines denote the maximum margin hyper-plane.



(a) (b)

Figure 8. CVM (a) V.S. S-CVM (b) under $\varepsilon = 10^{-4}$

e. Application of SMEB in multi-class classification

We compare the results obtained with one-v.s.-all CVM (OVA-CVM), one-v.s.-one CVM (OVO-CVM), multi-CVM (M-CVM) and multi-CVM with smaller core sets (SM-CVM) we proposed. The grid search method based on cross-validation is chosen to determine the values of the best model parameters in Gaussian kernel function k(u,v)=exp(-p_1(u-v)'(u-v))+ p_2, such as p_1=1, p_2=0, penalty factor C=100.

e.i Data Sets in Experiments

1) Synthetic data sets: We generated six synthetic data sets, each of which is a 2 × 2 check-board data set, and the points follow the uniform distribution.

2) Benchmark data sets: The benchmark data sets we used are from the UCI machine learning repository (http://archive.ics.uci.edu/ml/), iris, glass and balance.

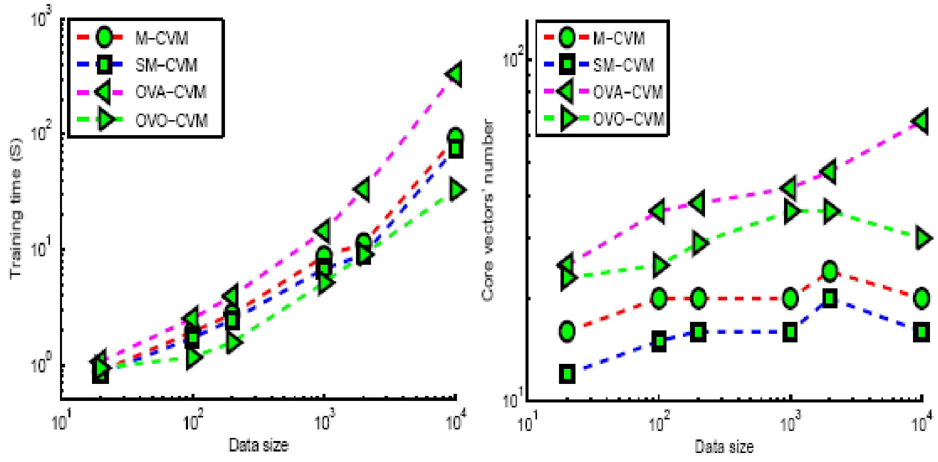
The details of data sets used in this section are listed in Table 7 below.

Table 7: Details of data sets

Data set	**Sy. 1**	**Sy. 2**	**Sy. 3**	**Sy. 4**	**Sy. 5**	**Sy. 6**	**Iris**	**Glass**	**Balance**
# Class	4	4	4	4	4	4	3	6	3
# Dim.	2	2	2	2	2	2	4	9	4
# Point	20	100	200	1000	2000	10000	150	214	625

e.ii Experimental results

We conduct the experiments with different ε on data sets mentioned before for all the algorithms to compare the performances. Training time and core vectors' number for all the algorithms, which vary with data size on the synthetic data under the best choice of ε are given in Figure 9. We can see that the proposed SM-CVM is of the smallest core vectors' number and the shortest training time, expect for the training time of OVO-CVM, which is of the lowest accuracy. Generally speaking, we can conclude that the proposed SM-CVM is of the shortest training time, the highest accuracy and the smallest core vectors' number.



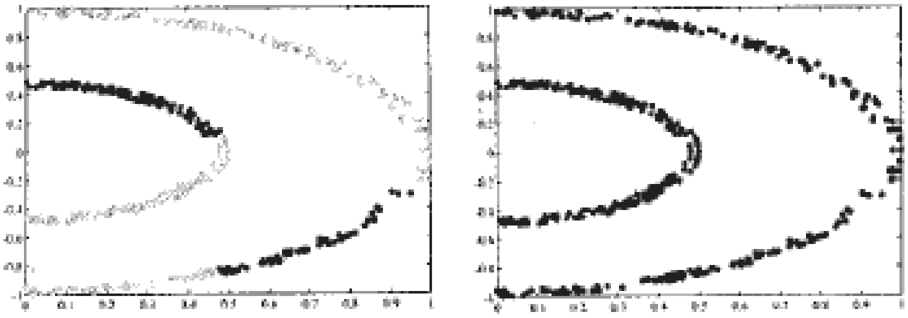
(a) (b)

Figure 9. Training time vary with data size (a), and core vectors' number vary with data size (b) for different algorithms

f. Application of SMEB in image clustering

In this section we utilize SMEB Algorithm in Table 1 to scale up image clustering problem, called SMEB-introduced Clustering.

Firstly, we tested the synthetic delta set (ftp.disi.unige.it/person/CamastraF/delta.dat) with the SMEB Algorithm. Delta set is formed by 424 training samples in R^2. The points are randomly distributed on two semicircles with the same center and 1inearly inseparable. Figure 10 below presents the initial status (a) and the final status (b) in the SMEB-introduced Clustering procedures for delta set, where we can see the points are clustered into two classes finally.



(a) (b)

Figure 10. Different status in SMEB-introduced Clustering procedures for delta set

Secondly, we scale up the delta sets to the size of $O(10^4)$, called the scaled-up delta sets. Figure 11 below shows the performance of SMEB-introduced Clustering on scaled-up delta sets in terms of both CPU time and the average error ratio, where we can see in Fig.11 (a) that the CPU time increases with the increasing of the size of the data set. For fixed degree of approximation, i.e., ε, the CPU time increases 1inearly in the size of the data set. Moreover, a larger ε corresponds to a shorter time. Fig.11 (b) shows that the average clustering error increases slowly as a function of ε, which means that the SMEB-introduced Clustering is a parameter non-sensitive algorithm for the degree of approximation ε.

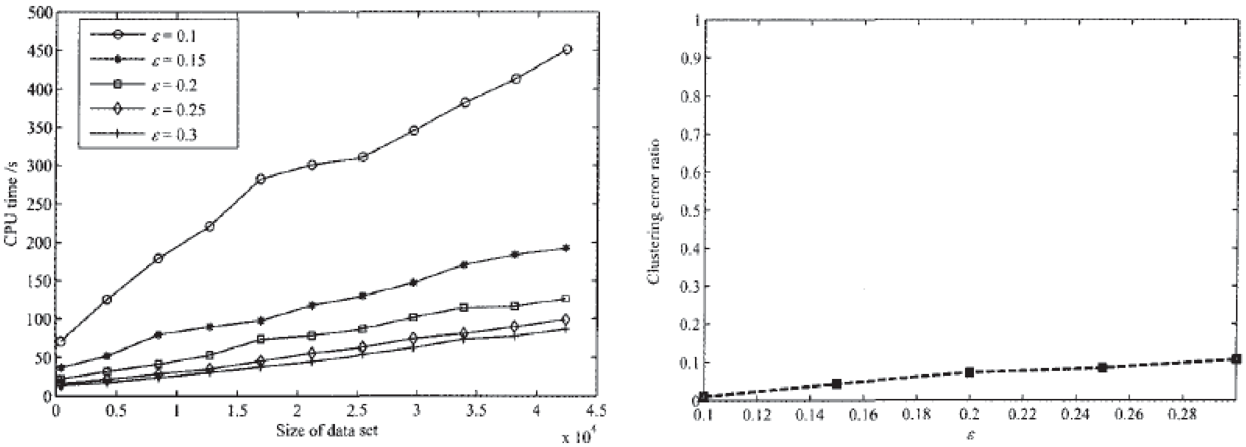


Figure 11. Experiment results on scaled-up delta sets using the SMEB-introduced Clustering

Thirdly, we used the SMEB-introduced Clustering to handle image segmentation based on color. The original images to be segmented are taken from the Berkeley image segmentation database (http://www.eecs.berkeley.edu/Research/Projects/cs/vision/grouping/segbench). Each image is a 48l×321×3 array of color pixels, where each color pixel is a triplet corresponding to red, green, and blue components. We explored the performance of our SMEB-introduced Clustering algorithm. The segmentation results were demonstrated in Fig. 12 and Fig. 13, where former presented the original images before being segmented, and later presented the segmented images. From the comparison we can see that the SMEB-introduced Clustering algorithm can achieve good performances in image clustering.

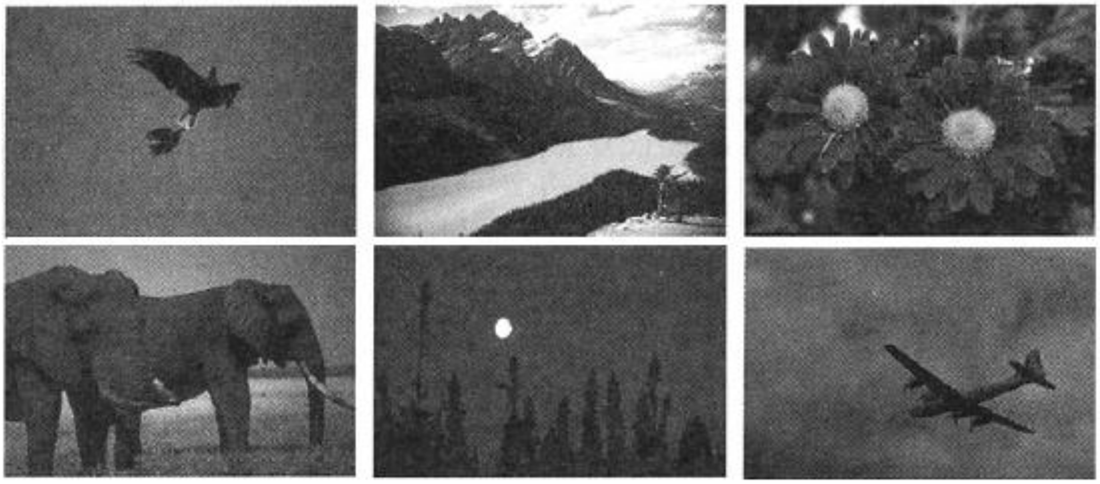


Figure 12: Original images before being segmented

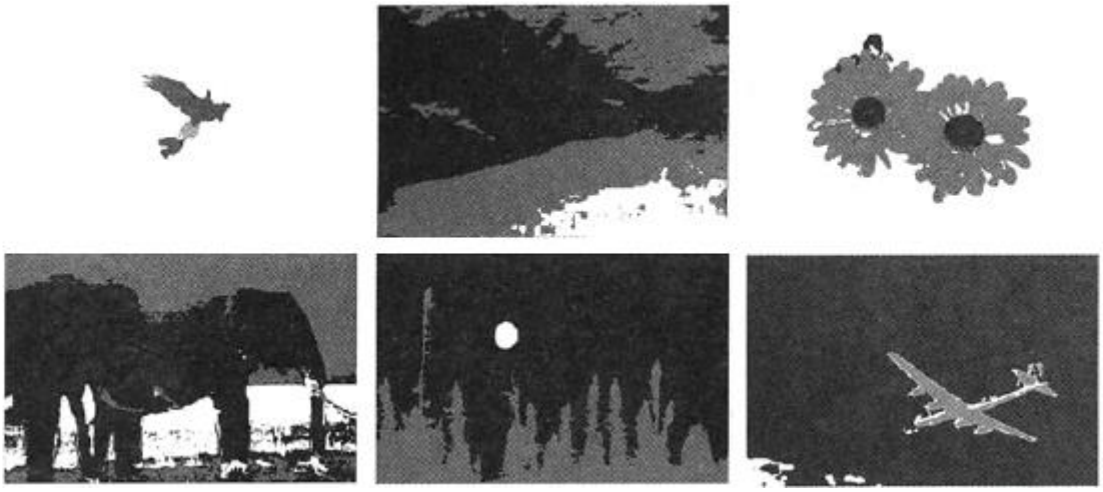


Figure 13: Experiment results on original images after being segmented

V. CONCLUSIONS

In this paper we develop a (1+ε)-approximate algorithm for computing the MEB of a given points set without requirement of any numerical solver. We prove theoretically that the proposed SMEB algorithm converges to the optimum within any precision in $O(1/\varepsilon)$ iterations. The SMEB has time complexity of $O(\frac{m}{\varepsilon^2}+\frac{1}{\varepsilon^3})$, which is linear in the number of training samples m for a fixed ε, and space complexity of $O(\frac{1}{\varepsilon^2})$, which is independent of m for a fixed ε. Compared to CVM and SCVM algorithms, it has the competitive performances in both training time and accuracy. Besides, the proposed algorithm does not need any extra requirement on kernels, which guarantees the potential applications in extensive kernel methods, consequently. We also use the proposed SMEB algorithm to handle Binary Classification, Multi-class Classification, and Image Clustering problems. Experiments on both synthetic and real-world data sets demonstrate the validity of the algorithm we proposed.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant No. 41001235), the Aeronautical Science Foundation of China (Grant No. 2011ZC55005), the Project of Henan Provincial Audit Office (Grant No. 20120922), the Scientific & Technological Research Key Project of Henan Provincial Education Office (Grant No. 13A520404), the Scientific & Technological Project of Henan Provincial Science & Technology Office (Grant No. 132102210468), and the Project of Science and Technology Bureau of Zhengzhou (Grant No. 20120435).

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES

[1] C. Cortes and V. Vapnik, “Support vector networks”, Machine Learning, Vol. 20, 1995, pp. 273-297.

[2] V. Vapnik, “Three Fundamental Concepts of the Capacity of Learning Machines”, Physica A, Vol. 200, 1993, pp. 537-544.

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

[4] R. Collobert, S. Bengio and Y. Bengio, “A Parallel Mixture of SVMs for Very Large Scale Problems”, Neural Computation, Vol. 14, No. 5, 2002, pp. 1105-1114.

[5] J. X. Dong, A. Krzyzak and C. Y. Suen, “A fast parallel optimization for training support vector machine”, Proc. of the 3rd Int. Conf. Machine Learning Data Mining, Vol. LNAI 2734, 2003, pp. 96-105.

[6] G. Zanghirati and L. Zanni, “A parallel solver for large quadratic programs in training support vector machines”, Parallel Comput., Vol. 29, No. 4, 2003, pp. 535-551.

[7] B. H. Guang, K. Z. Mao, C. K. Siew and D. S. Huang, “Fast modular network implementation for support vector machines”, IEEE Trans. Neural Netw., Vol. 16, No. 6, 2005, pp. 1651-1663.

[8] L. J. Cao, S. S. Keerthi and J. Q. Zhang, “Parallel Sequential Minimal Optimization for the Training of Support Vector Machines”, IEEE Trans. Neural Netw., Vol. 17, No. 4, 2006, pp. 1039-1049.

[9] S. Asharaf, M. N. Murty and S. K. Shevade, “Multiclass Core Vector Machine”, Proc. of ICML’07, 2007, pp. 41-48.

[10] Francesco Orabona, “Better Algorithms for Selective Sampling”, in Proceedings of ICML’11, 2011, pp. 433-440.

[11] Y. Liu, “Combining integrated sampling with SVM ensembles for learning from imbalanced datasets”, Information Processing and Management, Vol. 47, No. 4, 2011.

[12] M. Volpi, “Memory-Based Cluster Sampling for Remote Sensing Image Classification”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 50, No. 8, 2012.

[13] Abhimanyu Das and David Kempe, “Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection”, Proceedings of ICML’11, 2011, pp. 1057-1064.

[14] I. W. Tsang, J. T. Kwok and P. M. Cheung, “Core vector machines: Fast SVM training on very large data sets”, Journal of Machine Learning Research, Vol. 6, 2005, pp. 363-392.

[15] I. W. Tsang, A. Kocsor and J. T. Kwok, “Simpler Core vector machines with Enclosing Balls”, Proc. 24th Int. Conf. Machine Learning, 2007, pp. 911-918.

[16] K. Fischer, B. GÄartner and M. Kutz, “Fast smallest-enclosing-ball computation in high dimensions”, Proceedings of the 11th Annual European Symposium on Algorithms, 2003, pp. 630-641.

[17] P. Kumar, J. S. B. Mitchell and A. Yildirim, “Computing Core-Sets and Approximate Smallest Enclosing Hyper-Spheres in High Dimensions”, Proc. 5th Workshop on Algorithm Engineering and Experiments, 2003, pp. 45-55.

[18] D. M. J. Tax and R. P. W. Duin, “Support Vector Domain Description”, Pattern Recognition Letters, Vol. 20, No. 14, 1999, pp. 1191-1199.

[19] I. W. Tsang, J. T. Kwok and J. M. Zurada, “Generalizde Core vector machines”, IEEE Transactions on Neural Networks, Vol. 17, No. 5, 2006, pp. 1126-1140.

[20] Yongqing Wang, “Simpler Minimum Enclosing Ball: Fast Approximate MEB Algorithm for Extentive Kernel Methods”, CCDC’08, 2008, pp. 3492-3497.