Neurocomputing 172 (2016) 189–197

Contents lists available at ScienceDirect

Neurocomputing

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

Large-scale support vector machine classification with redundant data reduction

Xiang-Jun Shen [a,*], Lei Mu [a], Zhen Li [a], Hao-Xiang Wu [a], Jian-Ping Gou [a], Xin Chen [b]

[a] School of Computer Science and Communication Engineering, JiangSu University, JiangSu 212013, China
[b] Hermes Microvision Inc., San Jose, CA, USA

ARTICLE INFO

Article history:
Received 22 October 2013
Received in revised form
12 July 2014
Accepted 27 October 2014
Available online 8 May 2015

Keywords:
Support vector machine
Classification
Clustering
Redundant data reduction

ABSTRACT

Large-scale image classification has shown great importance in object recognition and image retrieval as the vast amounts of social multimedia sharing on the networks. While the time and memory requirements for SVM training surge with an increase in the sample size, which makes SVM impractical even for a moderate problem as the number of training data reaches to the extent of hundreds of thousands. To solve this problem, many specially designed algorithms are proposed such as clustering-based SVM training which attempts to remove the clustered data points that lie far away from support vectors. In this paper, we further explore that there exist clustered and scattered data points in a cluster. The clustered data points that lie around the clustering centroid are the dense data points, which are in the inner layer of a cluster. Those data points are viewed as having no SVs and removed. While the scattered data points are the sparse data points in the outside layer of a cluster. Those data points are viewed as having SVs and thus reserved. The Fisher Discriminant Ratio is employed to determine a boundary between the clustered and scattered data points in one cluster, which is computed based on the distance densities of data points to the cluster centroid. The redundant clustered data points in clusters are thus removed to speed up SVM training process. Several experimental results show that our proposed method has good classification accuracy while training time is significantly reduced. The training time in our proposed method only accounts for about 17 percent of the time in LIBSVM on the large data set of Covertype.

1. Introduction

Recent years have witnessed an explosion in the amount of images and videos in social media sharing websites, such as Flickr and Youtube, due to the spread of digital cameras, mobile devices and networking technology. An urgent need is how to effectively search these huge amounts of data efficiently. And this need has been recognized in the computer vision and machine learning research communities and large-scale classification methods have become an active topic of research [1–3] in recent years. Many classifiers, such as Nearest neighbor classifiers, decision tree, Bayesian classifiers, support vector machine (SVM), and the ensemble methods, have been proposed for image classification [4]. Among these technologies, SVM is the most prominent technology proposed by Vapnik [5] for its solid mathematical basis. It improves the generalization ability of a classifier by maximizing the margin between the two classes and thus achieves better classification accuracy on test data than using other classifiers on most popular benchmark data sets [6].

Despite its good theoretical foundations, SVM is not applicable for classification of large data set owing to vast time to train these large data set to obtain support vectors (SVs). And the SVs are the data points that are closest to the separating hyperplane in the training set. The decision for new data to be classified is solely based on the SVs. In order to obtain SVs, it is necessary to solve the quadratic programming (QP) problem, which depends heavily on the cardinality of the training set. And the optimization problem with big data causes an intensive computational complexity due to an increase in the number of data points used for training. This constraint makes SVM impractical even for a moderate problem [7] as the number of training data reaches to the extent of hundreds of thousands.

Much effort is devoted to reducing the time and space complexities when training large data sets. In general, these algorithms are divided into three categories: (I) dividing the original QP problem in SVM training into smaller QP sub-problems, (II) selecting a small number of representative training samples from the large data set to reduce the number of training data points, and (III) developing paralleled approaches which divide a large

* Corresponding author.
E-mail address: xjshen@ujs.edu.cn (X.-J. Shen).

http://dx.doi.org/10.1016/j.neucom.2014.10.102
0925-2312/

data set into smaller one, each independently running on separate computer nodes.

For the first class of algorithms, it includes chunking and decomposition methods, which are discussed by Osuna et al. [8], Boser et al. [9] and Kaufman [10]. Platt [11] proposes the Sequential Minimal Optimization (SMO) algorithm that transforms the large QP problem into a series of small QP problems, each of which optimizes only a subset of size two. Platt's SMO algorithm is further accelerated by Pavlov et al. [12] using Boost-SMO algorithm.

The second class of SVM training approaches is to choose representative samples to scale down the entire training data before SVM training. Lee and Mangasarian [13] propose the Reduced SVM (RSVM) that uses the random sampling technique to select a random subset of training data. Systematic Sampling RSVM (SSRSVM) [14,15] is further proposed to select the informative data points to form the reduced set. Active learning is another technique used in SVM training to reduce the number of training data points [16–18]. And Tsang [19] proposes core vector machine algorithm that samples the data points on Minimum enclosing ball. Meanwhile, another big sub-class of such algorithms is clustering. Clustering [20] is an unsupervised learning technique that classifies similar objects into groups (clusters), according to some criteria such as distance metrics. Clustering algorithms aim to removing the training data that form clusters far away from a separating hyperplane. For example, hierarchical clustering [21,22], adaptive clustering [23] and fuzzy clustering [24] techniques are used to decrease the complexity of SVM training. Cervantes et al. [25] further presents a minimum enclosing ball clustering algorithm to classify large data. And the clustering and RSVM techniques [26,27] are combined. As for the third class of distributed and parallel algorithms [28–33], they utilize the computing and storage capacities in distributed or parallel nodes and thus can complete the SVM training task that cannot be completed in one node.

Consider a typical two-class SVM training. As the SVs lie on the boundary of the convex hulls of binary classes, the data points that are far away from the hyperplane are not useful for classification and thus can be removed from the training data set. So the aim of such clustering-based SVM training algorithms is to remove the clusters that are far away from the hyperplane.

While compared with the work of clustering-based SVM training algorithms mentioned above, the novelty in our work is that we consider not only to remove the redundant clusters, but we further consider data distributions in clusters and the redundant data points in every cluster are removed. Thus we explore that the data points in every cluster can be classified as clustered and scattered data points. The clustered data points that lie around the clustering centroid are the dense data points, which are in the inner layer of a cluster. Those data points are viewed as having no SVs and removed. While the scattered data points are the sparse data points in the outside layer of a cluster. Those data points are viewed as having SVs and thus reserved. So our aim is to find boundaries between the clustered and scattered data points in clusters. In this paper, the Fisher Discriminant Ratio (FDR) criterion [4] is applied to find such boundaries in every cluster based on the distance densities of data points to cluster centroids. The redundant clustered data points in clusters are thus removed and finally speed the SVM training process much more. Several experimental results on simulated and real data sets show that, compared with LIBSVM[34], our proposed method keep good classification accuracy while the training time is significantly reduced. The training time in our proposed method only accounts for about 17 percent of the time in LIBSVM on the large data set of Covertype.

The remainder of this paper is organized as follows: Section 2 briefly introduces the optimization problem involved in SVM and two-class classification in Fisher Discriminant Analysis (FDA). Section 3 describes the proposed method based on data redundancy reduction. In Section 4, our experimental results are reported on both the simulated and real data sets. Section 5 gives a conclusion.

2. Support vector machine and Fisher discriminant analysis

In this section, the fundamental concepts of SVM and FDA are described briefly.

2.1. Support vector machine

Assume that a training data set of binary classes is $\mathbf{D}\{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{1,-1\}\}$, $i=1,2,\ldots,n$. $\mathbf{x}_i$ is the vector data point, n is the number of data points in the set $\mathbf{D}$ and the dimension of data point $\mathbf{x}_i$ is $\mathbb{R}^d$. y_i is the class membership of $\mathbf{x}_i$. Training SVM is to find the maximum margin hyperplane that separates the training data set.

The hyperplane is determined by a vector $\mathbf{w}$ with minimal norm and an offset b. A quadratic problem [5] needs to be resolved to find such an optimal hyperplane:

$$\min_{\mathbf{w},\boldsymbol{\xi}} : G(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

subject to

$$y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b) \geq 1-\xi_i, \quad \xi_i \geq 0$$

where ξ_i is a slack variable to tolerate mis-classifications. C is a parameter that determines the cost of the constraint violation. The variable b determines the offset of the hyperplane from the origin. $\phi(\mathbf{x}_i) = (\mathbf{x}_i^1, \ldots, \mathbf{x}_i^m)$ is a mapping from vector field $\mathbb{R}^d$ into feature space $\mathbb{R}^m$ which is a higher dimension Hilbert space H. And $\langle\cdot,\cdot\rangle$ denotes the dot product in H. $\phi(\cdot)$ is a nonlinear function by using a kernel $K(\cdot,\cdot)$. And the kernel must satisfy the Mercer condition [5]: $K(\mathbf{x}_i,\mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$.

With the help of Lagrangian multipliers [5], the dual form of the above minimization problem in Eq. (1) is equivalent to

$$\max_{\boldsymbol{\alpha}} : W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{n} y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)\alpha_i\alpha_j + \sum_{i=1}^{n}\alpha_i \tag{2}$$

subject to

$$\sum_{i=1}^{n}\alpha_i y_k = 0, \quad 0 \leq \alpha_i \leq C$$

where $\boldsymbol{\alpha}$ is a vector with components α_i that is the Lagrange multiplier.

The necessary and sufficient conditions for a weight vector $\mathbf{w}$ and Lagrange multipliers $\boldsymbol{\alpha}$ to optimal are the KKT conditions [5]. Many solutions of Eq. (2) are zero, which mean most α_i are zero, Based on the non-zero α_i, the optimal vector $\mathbf{w}$ is:

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \phi(\mathbf{x}_i) \tag{3}$$

where $\mathbf{w}$ is expressed by a linear combination of support vectors (SVs). The elements in the set SVs are the subset of training sample vectors $\{\mathbf{x}_i\}_{i=1}^{l}$, which have the l non-zero Lagrange multipliers $\{\alpha_i\}_{i=1}^{l}$. And the resulting classifier is

$$y(\mathbf{x}) = \sum_{i\in SVs}\alpha_i y_i K(\mathbf{x}_i,\mathbf{x}) + b,$$
$$\text{sign}(y(\mathbf{x})) = \begin{cases} +1 : y(\mathbf{x}) > 0 \\ -1 : y(\mathbf{x}) < 0 \end{cases} \tag{4}$$

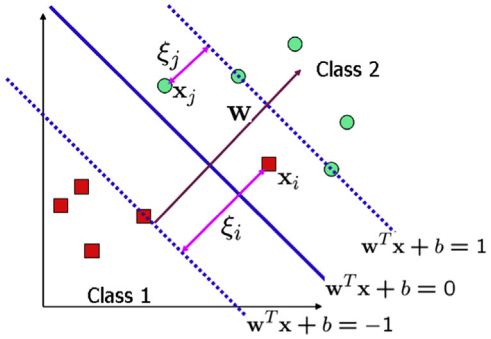


Fig. 1. Examples of SVM classifier. The decision boundary is $\mathbf{w}^T\mathbf{x}+b=0$ and the support vectors are the data points on the dot lines.

The distance between a data point $\mathbf{x}$ and the decision hyperplane is

$$dist=\frac{\|y(\mathbf{x})\|}{\|\mathbf{w}\|} \tag{5}$$

where $\|\cdot\|$ is a distance metric and usually is the Euclidean distance. For example, $\sqrt{\sum_{j=1}^{d}(\mathbf{w}_j)^2}$ is the Euclidean distance between $\mathbf{w}$ and 0. $\mathbf{w}_j$ is the jth component of $\mathbf{w}$.

From Eq. (4), only the set of support vectors can determine the decision boundary. And the other training data that is not the support vectors can be removed without influencing the final decision boundary. An example of resulting hyperplane is shown in Fig. 1.

2.2. Fisher discriminant analysis

The objective of FDA is to seek a direction that can best separate the two classes [4]. To find such a direction, suppose the set $\mathbf{D}$ is divided into two subsets $\mathbf{D}_1$ and $\mathbf{D}_2$ according to their class labels. The goal is to find a projection onto a line y as

$$y=\mathbf{w}_\mathbf{d}^T\mathbf{x} \tag{6}$$

where the data points $\mathbf{x}$ corresponding to $\mathbf{D}_1$ and $\mathbf{D}_2$ are well separated by the direction of $\mathbf{w}_\mathbf{d}$. The criterion function for the best separation is defined as

$$J(\mathbf{w}_\mathbf{d})=\frac{\mathbf{w}_\mathbf{d}^T\mathbf{S}_B\mathbf{w}_\mathbf{d}}{\mathbf{w}_\mathbf{d}^T\mathbf{S}_W\mathbf{w}_\mathbf{d}} \tag{7}$$

$$\mathbf{S}_B=(\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^T \tag{8}$$

$$\mathbf{S}_W=\mathbf{S}_1+\mathbf{S}_2 \tag{9}$$

where the sample mean is $\mathbf{m}_i=(1/|\mathbf{D}_i|)\sum\mathbf{x}\in\mathbf{D}_i\mathbf{x}$, and $\mathbf{S}_i=\sum\mathbf{x}\in\mathbf{D}_i(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T$ is the sample covariance. $i=1,2$. $\mathbf{S}_B$ denotes the within-class scatter matrix and $\mathbf{S}_W$ denotes the between-class scatter matrix. The function $J(\mathbf{w})$ tries to find a projection that can make the difference between the means of two classes as large as possible relative to their covariances. And the optimal $\mathbf{w}_\mathbf{d}$ can be computed as

$$\mathbf{w}_\mathbf{d}=\mathbf{S}_W^{-1}(\mathbf{m}_1-\mathbf{m}_2) \tag{10}$$

Fig. 2 gives an example of projection data samples on Fisher Discriminant Analysis. The direction of $\mathbf{w}_1$ is computed by FDA and the direction of $\mathbf{w}_2$ is selected randomly. From the figure, the projection in the direction of $\mathbf{w}_1$ shows greater separation between the square and circle data points than in the direction of $\mathbf{w}_2$.

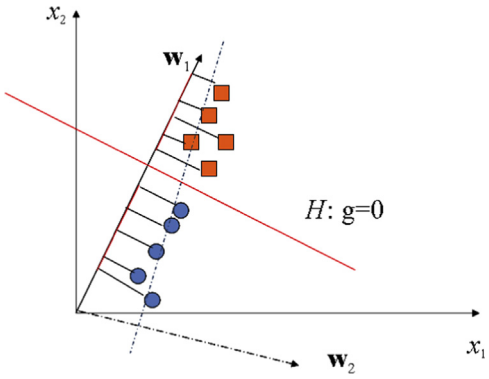


Fig. 2. Projection of data samples onto two different lines in the directions marked as $\mathbf{w}$. The projection in the direction of $\mathbf{w}_1$ shows greater separation between the square and circle data points than in the direction of $\mathbf{w}_2$.

This criterion takes a special form in the one-dimensional. The so-called Fisher Discriminant Ratio (FDR) result is

$$F=\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2} \tag{11}$$

where μ_1 and μ_2 denote the mean of two classes, σ_1 and σ_2 are their standard variance.

3. Support vector machine training with redundant data reduction

After introducing the concepts of SVM and FDA, the proposed SVM training with redundant data reduction is introduced in this section. Firstly, redundant data reduction through boundary finding in one cluster is introduced. Then the algorithms of our proposed method are explained in the following subsection.

3.1. Redundant data reduction through boundary finding in one cluster

Assume there are n data points $\{\mathbf{x}_i\}_{i=1}^n$ in one cluster. The dimension of data point $\mathbf{x}_i$ is $\mathbb{R}^d$. The current centroid of data is $\boldsymbol{\mu}=(1/n)\sum_{i=1}^n\mathbf{x}_i$. The n data points are sorted in ascending order according to their distances to the centroid $\boldsymbol{\mu}$. For $\mathbf{x}_i$, the distance is $\|\mathbf{x}_i-\boldsymbol{\mu}\|$ and the distance metric used in this paper is the Euclidean distance. The sorted distance set is denoted by $\{d_i, i=1,2,\ldots,n\}$ and the corresponding data point set according to this sorted distance set is denoted by $\{\mathbf{sx}_i, i=1,2,\ldots,n\}$. d_i is the ith shortest distance to the centroid $\boldsymbol{\mu}$ and the corresponding data point is $\mathbf{sx}_i$. That means $\mathbf{sx}_1$ is the nearest data point, while $\mathbf{sx}_n$ is the farthest one.

We further define a distance density set *Den*. The element in the set is defined as

$$Den_i=\frac{i}{\pi d_i^2} \tag{12}$$

where Den_i is the distance density that counts the data points fall in the circle of $(\boldsymbol{\mu}, d_i)$, where the center is located at $\boldsymbol{\mu}$ and the radius is d_i. Den_1 only counts the nearest data point $\mathbf{sx}_1$ within the distance d_i and Den_n counts all the data points within the distance d_n.

Obeying a general assumption, the data points near the centroid are assumed clustered and the data points away the centroid are assumed scattered. So the distance densities of clustered data points are dense and the distance densities of scattered data points are sparse. A boundary between the clustered and scattered data points is thus found through the distance

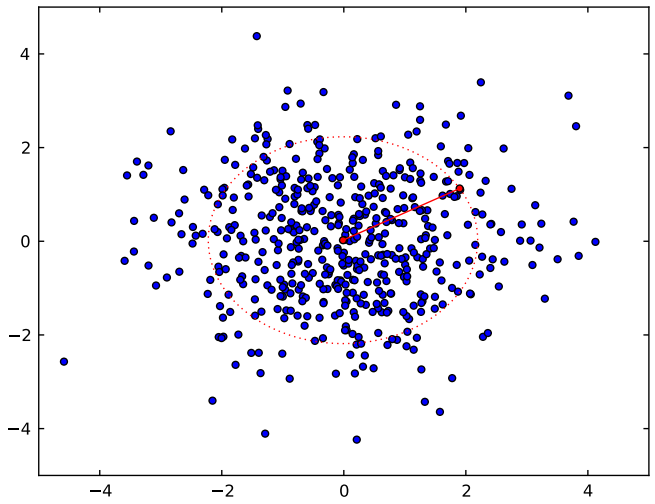


Fig. 3. Boundary learned dynamically from a toy sample data to separate between the clustered and scattered data points. The data points within the boundary are the clustered data points and are removed from the training set. Only the scattered data points that are out of the boundary are reserved for SVM training.

densities of data points. To find such a boundary that optimally partitions the clustered and scattered data points, suppose that the rth data point $\mathbf{sx}_r$ is selected as the boundary. Accordingly, the n data points are partitioned into the clustered and scattered set as $X_c = \{\mathbf{sx}_1, \ldots, \mathbf{sx}_r\}$ and $X_s = \{\mathbf{sx}_{r+1}, \ldots, \mathbf{sx}_n\}$. The corresponding distance sets of X_c and X_s are $D_c = \{d_1, \ldots, d_r\}$ and $D_s = \{d_{r+1}, \ldots, d_n\}$, respectively. Based on the Fisher Discriminant Ratio defined in Eq. (11), we can find the optimal boundary by solving the following optimization problem:

$$F(r) := \frac{(\mu_c - \mu_s)^2}{\sigma_c^2 + \sigma_s^2}$$

$$J = \underset{r \in \{1,2,\ldots,n-1\}}{\operatorname{argmax}} F(r) \tag{13}$$

where $F(r)$ denotes the Fisher Discriminant Ratio between X_c and X_s. μ_c and μ_s denote the mean distance of D_c and D_s, σ_c^2 and σ_s^2 are the distance variance in D_c and D_s, respectively. The data point that leads to the maximal Fisher Discriminant Ratio J is regarded as the optimal boundary.

As shown in Fig. 3, the data points within the boundary are the clustered data points and are removed from the training set. Only the scattered data points that are out of the boundary are reserved for SVM training.

3.2. SVM training with redundant data reduction

Applying the idea of redundant data reduction through boundaries finding in clusters, our proposed method uses the following two steps to fulfill the training task: at the first stage, redundant clusters are removed. A clustering algorithm, i.e. K-Means [4], is first applied to obtain clusters. The clusters then are further divided into smaller clusters according to the class labels of data points. An approximate hyperplane is then obtained through using the centroids of clusters as training data. And the MMCD (Max–Min Cluster Distance) algorithm is used to remove the redundant clusters that are far away from the approximate hyperplane. At the second stage, clustered data points in every remaining cluster are removed through FIFDR (Fast Iteration of FDR) algorithm, which uses boundaries finding in the Section 3.1. After the clustered data points are removed, the reserved data points are fed into SVM training algorithm such as SMO algorithm. [34].

3.2.1. Removing redundant clusters

In order to remove the clustered data points in clusters, we need to cluster the original training data points into several clusters firstly. Suppose the training data set $\mathbf{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{1, 2, \ldots, M\}\}, i = 1, 2, \ldots, n$. And assume the set has M different class labels.

The K-Means algorithm [4] is applied to cluster the training data into K clusters. i.e. $C = \{c_i | i = 1, \ldots K\}$. The parameter K is preset and effected subjectively by users' domain knowledge. While the clustering method used in our work is to divide the whole data set into smaller groups. A bigger value of K is preferred. The K clusters are further classified into two subsets of $UC = \{uc_i | i = 1, \ldots R. 1 \le r \le K\}$ and $MC\{mc_j | j = 1, \ldots K - R\}$, respectively. $C = UC \cup MC$. In every cluster uc_i, the data points have one single class label. While in every cluster mc_j, the data points have two or more class labels. And every cluster in MC are further divided into sub-clusters where the data points have one single class label. For example, in cluster mc_j, there are l class labels among the data. The clusters are then divided into l sub-clusters. Assume MC is further divided into L sub-clusters $UMC = \{umc_j | i = 1, \ldots L\}$. Finally, the set C is divided into $R+L$ sub-clusters. And the $R+L$ sub-cluster is named $S = UC \cup UMC = \{uc_1, \ldots, uc_R, umc_1, \ldots umc_L\}$.

As the SVs lie on the boundary of the convex hulls of two or more classes, there is a higher probability that SVs are in the clusters of MC. Meanwhile, there is a lower probability that SVs are in the clusters of UC for the clusters have only one single class label, which means that the clusters of UC are farther away from the hyperplane than the clusters of MC. To remove the redundant clusters in UC, assume the centroid set $A = (\mu_i, y_i) | y_i \in \{1, 2, \ldots, M\}$ is input as the training data. $i = 1, \ldots, R+L$. And the centroid $\mu_i = (1/n_i) \sum_{i=1}^{n_i} \mathbf{x}_i$ is the cluster i of set S. The normal SMO algorithm such as LIBSVM [34] is used to obtain the approximate decision hyperplane defined in Eq. (4). As there are M class labels and SVMs can only solve binary classification problems, the one-against-one technique is used and the number of decision hyperplane is $M(M-1)/2$. Then the Max–Min Cluster Distance (MMCD) algorithm is applied and detailed in Algorithm 1. Applying the algorithm, a reduced set RS_1 is finally output.

Algorithm 1. Max–Min cluster distance algorithm.

1: $i \leftarrow 0$.
2: **while** $(i < M)$ **do**
3: **for** (each cluster umc_k in UMC whose class label y_k is i) **do**
4: **for** (each data point j in umc_k) **do**
5: Compute the distances defined in Eq. (5) between the data point j and the $M-1$ decision hyper-planes. Reserve the shortest distance as the distance of data point j.
6: **end for**
7: Assume the distance set of data points in cluster umc_k is DD_k. Reserve its largest distance among DD_k as the distance of cluster umc_k.
8: **end for**
9: After the distances of all clusters in UMC are computed, reserve the largest distance among those distances. Assume this distance is $CMax_i$.
10: **for** (each cluster uc_j in UC whose class label y_j is i) **do**
11: **for** (each data point m in uc_j) **do**
12: Compute the distances between the data point m and the $M-1$ decision hyper-planes. Reserve the shortest distance as the distance of data point m.
13: **end for**
14: Assume the distance set of data points in cluster uc_j is UD_j. Reserve its shortest distance among UD_j as the distance of cluster umc_k. And assume this cluster distance is $CMin_j$.

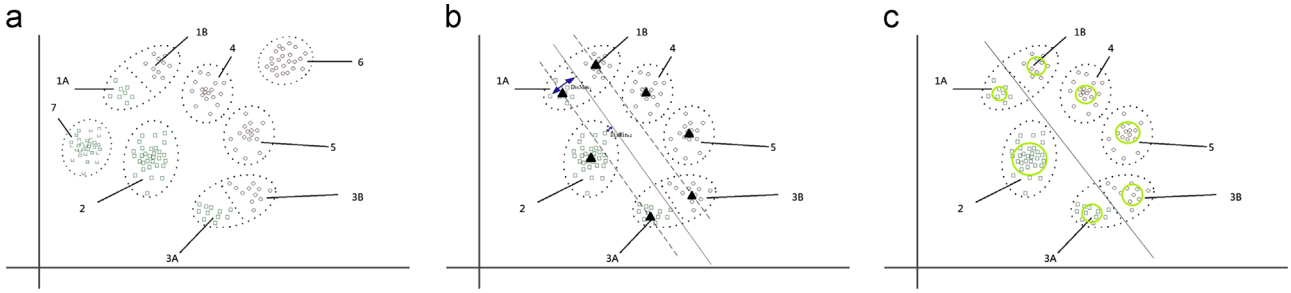


Fig. 4. Our proposed method of SVM training in a sample data set, (a) clustering results of sample data, (b) results of removing redundant clusters, (c) results of removing redundant data in every cluster.

15: **if** $(CMin_j > CMax_i)$ **then**
16: Remove the cluster uc_j.
17: **else**
18: Reserve the cluster uc_j.
19: **end if**
20: **end for**
21: $i \leftarrow i+1$
22: **end while**
23: Assume the reserved UC is $\{uc_m \mid m = 1, \ldots, H\}$. Output $\mathbf{RS}_1 = \{uc_1, \ldots, uc_H, umc_1, \ldots, umc_L\}$

Fig. 4 illustrates our method of SVM training in a sample data set. Fig. 4(a) and (b) shows the first stage of redundant clusters removing. Fig. 4(a) illustrates the clustering results of sample data points using K-Means algorithm. The data points have two class labels and the binary data points are denoted by square and circle, respectively. Six clusters are obtained and enclosed by dotted ellipse. Clusters 1 and 3 are further divided into 4 sub-clusters as 1A, 1B, 3A, 3B, according to their class labels. Then those 4 sub-clusters with the clusters 2, 4, 5, 6 and 7, are input to the MMCD algorithm. Fig. 4(b) illustrates the results of removing the redundant clusters by the MMCD algorithm. The approximate SVs are on the dotted lines and the solid line is the approximate decision hyperplane. From the figure, clusters 6 and 7 are removed and the remaining clusters are fed to the next stage to remove the clustered data points in every remaining cluster.

3.2.2. Removing clustered data points in every cluster

To remove the clustered data points in every cluster of RS_1, the idea of boundary finding in Section 3.1 is used in this subsection. Assuming there are Z data points in one cluster, the optimal boundary is obtained by calculating $Z-1$ values of FDR. And every time when the FDR value is calculated, the mean and variance distances in two classes are computed. For such huge data in every cluster of RS_1, it is time consumption. We thus propose a Fast Iteration of FDR (FIFDR) method to accelerate obtaining the optimal boundary in each cluster of RS_1. To obtain the optimal boundary quickly, different iterative step lengths are adopted to narrow down the range of distance densities, which are calculated through FDR method. In each iteration, the current range of distance densities to be calculated is narrowed and obtained by the previous iteration.

For the ith cluster rs_1^i of RS_1, assume there are z_i data points in cluster rs_1^i and the data points are $\{\mathbf{x}_i\}_{i=1}^{z_i}$. The data centroid is $\boldsymbol{\mu}_i$. The sorted distance set according to the centroid is denoted by $\{dist_i, i = 1, 2, \ldots, z_i\}$. The corresponding data point set according to the sorted distance set is denoted by $\{\mathbf{sx}_i, i = 1, 2, \ldots, z_i\}$ and the corresponding distance density set is denoted by $Den_i = \{den_i \mid i = 1, \ldots, z_i\}$, which is defined in the formula (12). Then the proposed FIFDR method in the cluster rs_1^i is detailed in Algorithm 2.

Algorithm 2. Fast iteration of FDR algorithm.

1: Set the iteration variable $IL=1$ and the maximum iteration variable $IM = mg+1$. Set $B_{min} = 1$ and $B_{max} = z_i$.
2: **while** $(IL < IM)$ **do**
3: Compute the current iterative step length $SL = \frac{B_{max} - B_{min} + 1}{(B_{max} - B_{min} + 1)^{1/(IM - IL)}}$
4: **for** $(r = B_{min};\ r <= B_{max};\ r += SL)$ **do**
5: Divide the data points into two sets as $X_c = \{\mathbf{sx}_1, \ldots, \mathbf{sx}_r\}$ and $X_s = \{\mathbf{sx}_{r+1}, \ldots, \mathbf{sx}_{z_i}\}$. And compute its FDR value F_r, according to the formula (11).
6: **end for**
7: Select the two largest FDR values among the $nv = (B_{max} - B_{min})/SL$ values. And assume $F(x)$ and $F(y)$ are the two largest values.
8: $B_{max} \leftarrow x$ and $B_{min} \leftarrow y$
9: $IL \leftarrow IL+1$
10: **end while**
11: Remove the data points $\{\mathbf{sx}_1, \mathbf{sx}_2, \ldots, \mathbf{sx}_{B_{max}}\}$. And the data point set $\{\mathbf{sx}_{B_{max}+1}, \mathbf{sx}_{B_{max}+2}, \ldots, \mathbf{sx}_{z_i}\}$ is reserved and named rs_2^i. Finally output the set rs_2^i.

In the algorithm, mg is the magnitude of the given data set. For example, if a data set has about 10^4 data points, mg is set 4 and the initial iterative step length SL is set $10^4/10 = 1000$. In the final iteration, the iterative step length SL is decreased to one and the optimal boundary is obtained finally. The parameters B_{min} and B_{min} are used to control the current range of distance densities to be calculated. The output data points are the set rs_2^i. And we thus apply the FIFDR algorithm to all the clusters of RS_1 and finally obtain the reduced data set denoted by $RS_2 = \{rs_2^i \mid i = 1, \ldots, H+L\}$.

Fig. 4(c) illustrates our proposed FIFDR method in a sample data set to remove the clustered data points in remaining clusters of 1A, 1B, 3A, 3B, 2, 4 and 5. The clustered data points in every cluster are enclosed by solid circle and removed. While the data points that lie between the dotted ellipse and solid circle in every cluster are the scattered data points and finally reserved.

Finally the reduced data set RS_2 is feed into SVM for training. As the original data set is reduced significantly, the time and space requirements of training are vast reduced in comparison with the original data set.

3.2.3. Time and space complexities

For comparison of time and space complexity in our propose method and normal SVM, the time and space complexities of normal SVM are considered first. The training time of SVM is dominated by the time for solving the underlying QP, and so the theoretical and empirical complexity varies depending on the

method used to solve it. The standard result for solving QP is that it has $O(n^3)$ time and $O(n^2)$ space complexities [19].

As for our proposed method, the time and space complexities of our method are then divided into 4 parts. The first part is the time and space complexities of K-Means clustering algorithm. Then the second part is the complexities of our MMCD algorithm. The third part is the complexities of our FIFDR algorithm. The last part is the time and space complexities of the normal SVM. In this part only the remaining data set is input to the normal SVM.

For the first part, most of the time is spent on computing vector distances. The clustering can be solved in time [35] $O(n^{dK+1}\lg n)$, where n is the number of data points to be clustered and d is the number of feature dimensions. The space requirement in such algorithm is $O(I\times K\times d\times n)$, where I is the number of iterations and $I\times K\times d\ll n$.

For the second part of MMCD in our proposed method, the time is spent on computing the distances from the data points in the training set to the approximate decision hyperplane. This distance is defined in Eq. (5) and its time complexity is $O(l\times m)$, where l is the number of SVs and m is the number of Hilbert space dimensions that are mapped from vector field $\mathbb{R}^d$ into feature space $\mathbb{R}^m$. For the n data points in the training set, our MMCD time complexity is $O(n\times l\times m)$, $(n\times l\times m\times n^3)$. And its space requirement is $O(l\times m)$, $(l\times m\times n^2)$.

For the third part of FIFDR in our proposed method, the time is spent on finding the maximum FDR values in every remaining cluster. Assume there are z_i data points in one cluster, sorting algorithm is firstly used such as quick-sort algorithm in this part. And the time and space complexities in such algorithm are $O(z_i\lg z_i)$ and $O(\lg z_i)$. Then the time and space complexities in FIFDR algorithm are to find the maximum FDR values which is determined by the step length and the number of iterations. The time and space complexities in computing the values of FDR are $O(z_i\times z_i/MSL)$ and $O(z_i)$, where MSL is the first step length computed in FIFDR algorithm.

Finally, for the last part, we use the normal SVM algorithm. And the time and space complexities in this part are depended on the size of remaining data points. Less the remaining data points are, less the time and space complexities need.

From the above analysis, it demonstrates that our proposed method has a lower time and space requirements than the normal SVM algorithm. And in the next section, we demonstrate them on several simulated and real data sets.

4. Experimental results

To verify the effectiveness of our proposed method for large data set classification, 1 simulated data set and 3 real-world data sets are used. Several experiments are designed and performed on these data sets along with the LIBSVM toolbox that implements the normal SMO algorithm. And the clustering-based SVM algorithm [24] is used for comparison. The classification accuracy and training time are used to measure the performance of those classifiers. The 3 classification methods are implemented by Python. All experiments are run on a Core i3-2120 3.3 GHz CPU with 2GB memory. The kernel function used in our proposed method and LIBSVM are Gaussian RBF function $K(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\gamma\times\|\mathbf{x}_i-\mathbf{x}_j\|^2)$. The Gaussian kernel is used with parameters $\gamma=1.0$. And the penalty parameter $C=10$. The 4 data sets are described as follows.

Simulated data set: A binary-class data set which has 50,000 two-dimensional data points. The training data points are sampled independently from a mixture of three Gaussian distributions with means centered at $(1,-1)$ and $(5,1)$. And the covariance matrices are $\left(\begin{smallmatrix}3 & 3\\ 0 & 1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}4 & 1\\ 0 & 1\end{smallmatrix}\right)$. The test data points are constructed in the same way and its data size is 50,000.

Adult data set: This data set is from the UCI machine learning database repository [36]. The database has nine groups (Adult-1a, Adult-2a,…,Adult-9a) of training and testing data. The goal of this data mining task is to predict whether a household has an income greater than $\$50,000$. The Adult-9a group contains 32,561 training data and 16,281 testing data. Each datum has 123 attributes.

Covertype data set: This data set is also from the UCI machine learning database repository. The original UCI Covertype data consists of 7 classes of trees, where the inputs are terrain features. There are 581,012 data samples with 54 attributes. The data set is preprocessed and transformed from multi-class into binary-class. 570,000 samples are used for training and the remaining 11,012 samples for testing.

Pascal VOC2012 data set: This data set is distributed for the task of the social object recognition and annotation [37]. The data consists 17,000+ images of 20 classes and 10,000 images is used for removing some semantic ambiguous images. The SIFT features [38] and the BoW model [39] are used to represent images. More than $1M$ SIFT features are extracted and the vocabulary size is 1,000. Thus the training data samples are 10,000 with 1,000 attributes. The testing data set is build the same way and its size is 2,000. The data set is multi-class and the number of decision hyperplane is 190.

Firstly, the classification effect and training time of the number of initial clusters K is studied through the 3 data sets, which is shown in Fig. 5. The values of K vary from 2 to 150 with an interval of 5. From Fig. 5(a), (c) and (e), the figures show that we achieve better classification results than the clustering based algorithm. And from Fig. 5(c), (d) and (f), they illustrate that the training time increases with the values of K increase. This is because the clustering algorithm needs more time with the values of K increase. To trade off the classification accuracy and the training complexity, the value of K is set according to the size of training data set and the number of real class labels. In our experiments, the value of K is set as 20–30 times to the number of class labels of a data set.

Secondly, the scale performance of our proposed method with LIBSVM and clustering-based SVM algorithm [24] on the Covertype data set is compared, which is shown in the Fig. 6. The number of training data varies from 50,000 to 550,000 with an interval of 50,000 and the value of k is set 100. A similar classification accuracy is achieved with the LIBSVM, which is shown in Fig. 6(a). And Fig. 6(b) shows that, with the number of training data being larger, the training time becomes longer in 3 classifiers. While the normal SMO algorithm implemented in the LIBSVM toolbox removes no training data, it shows worst training time among the 3 classifiers. And our proposed method achieves slowest growth rate of time among the 3 methods, for our proposed method removes much more redundant training data points than the clustering-based algorithm. From the figure, it demonstrates that our proposed method can achieve best training time efficiency among the 3 methods, especially in a massive data set.

Thirdly, we compare the training time and classification accuracy of our proposed method with LIBSVM and clustering-based classification method on the 4 data sets with the value of K being fixed. The values of K set in the simulated data set, Adult-9a, Covertype and Pascal VOC2012 data set are 80, 60, 100 and 200 respectively, according to its data size and class labels. And the time efficiency and classification accuracy are seen in Table 1. From the table, it has clearly shown that the classification accuracy (see Acc column) in our proposed method is better than the clustering-based classification method. While considering the training time (see time column) in the three algorithms, our

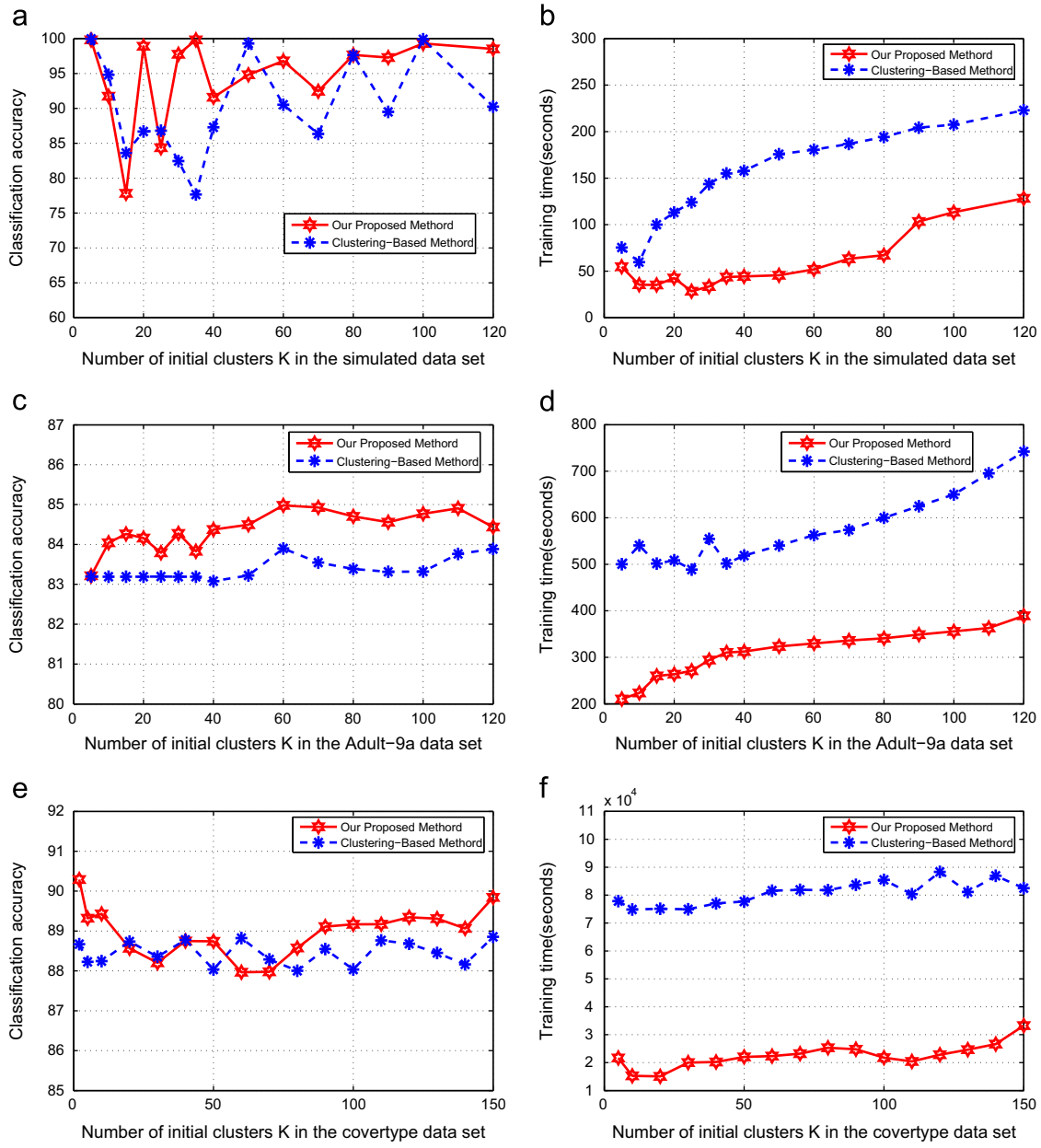


Fig. 5. Classification accuracy versus training time as a function of the number of initial clusters k in 3 data sets. The number of clusters k varies from 2 to 150 with an interval of 5: (a) classification accuracy of the simulated data set, (b) training time of the simulated data set, (c) classification accuracy of the adult data set, (d) training time of the adult data set, (e) classification accuracy of the Covertype data set, (f) training time of the Covertype data set.

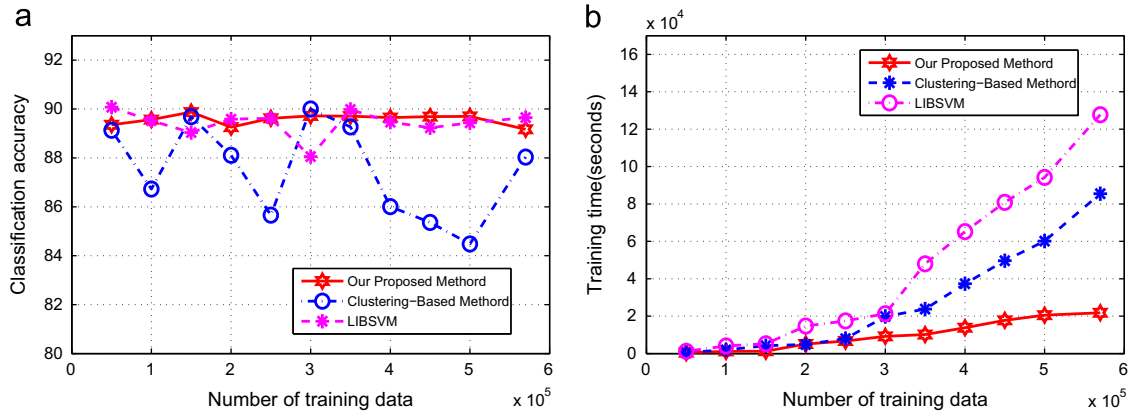


Fig. 6. Comparison of the scaling performance of our proposed method, clustering-based method and that of LIBSVM on the Covertype data set. The number of training data varied from 50,000 to 550,000 with an interval of 50,000: (a) comparison of classification accuracy in the 3 methods, (b) comparison of training time in the 3 methods.

Table 1
The performance comparison in terms of the classification accuracy and time efficiency with the initial clusters K being fixed.

Data set	Our proposed method			Clustering-based method			LIBSVM		
	Acc	Time	Reserved	Acc	Time	Reserved	Acc	Time	Original data points
Simulated data	97.66	67.09	29,482	97.64	194.19	42,152	99.01	107.29	50,000
Adult-9a	84.98	329.76	17,050	83.90	562.60	24,800	84.99	693.49	32,561
Covertype	89.17	21,762.67	334,891	88.03	85,461.77	486,589	89.65	127,745.92	570,000
Pascal VOC2012	46.64	507.15	6,658	45.69	713.10	8,164	47.12	895.68	10,000

Table 2
Classification accuracy on 10 representative image categories on Pascal VOC2012 data set.

Algorithms	Data set									
	Aeroplane	Bicycle	Bird	Boat	Bottle	Car	Cow	Horse	Person	Train
Our proposed method	66.16	56.21	44.38	50.89	32.54	68.05	46.15	37.87	47.93	56.21
Clustering-based method	63.01	68.21	35.84	40.46	39.88	61.85	32.37	38.73	42.20	53.76
LIBSVM	66.27	65.68	41.42	40.83	29.59	61.54	46.15	46.75	45.56	50.88

proposed method shows best time performance among the 3 methods, especially in the large data set of Covertype. And the training time in our proposed method only accounts for about 17 percent of the time ($21762.67/127745.92 = 0.17$) in LIBSVM on the large data set of Covertype. This is because our proposed method reserved (see Reserved column) lest data points among the 3 method. And since LIBSVM does not remove any data points, it shows the worst training time performance among the 3 algorithms.

Table 1 gives an average classification accuracy on Pascal VOC2012 data set. And Table 2 shows detailed classification accuracy of 3 algorithms on 10 representative image categories. From the table, it illustrates that our proposed method achieves better classification accuracy on image categories of aeroplane, bird, boat, car, cow, person and train. Also there is also worse classification accuracy observed in image categories of bicycle and horse.

From the above experiments, they illustrate that our proposed method can preserve the data points of being possible support vectors, while discard redundant data points that are useless in training. Classification accuracy is thus preserved and the training time is vastly reduced.

5. Conclusions

In this paper, we propose an effective SVM training method to remove the clustered data points in clusters successfully. Boundaries between the clustered and scattered data points are learned through maximizing value of FDR in clusters. Therefore the clustered data points in clusters are removed. The experiments conducted both on simulated and real data sets demonstrate that our approach significantly reduces the training time and preserves good classification accuracy at same time. On the large data set of Covertype, the training time in our proposed method only accounts for about 17 percent of the time in LIBSVM.

These results demonstrate the potential for classification of large-scale image collections on the social networks. For our future work, we are extending the classification benchmark for further parallel classification evaluations and investigating scalable methods on imbalanced large-scale image data sets for effective expansion.

Acknowledgements

This work was funded in part by the National Natural Science Foundation of China (No. 61005017); Natural Science Foundation of the Jiangsu Higher Education Institutions of China (10KJB520005); Senior Talent of Jiangsu University (No. 1283000347).

References

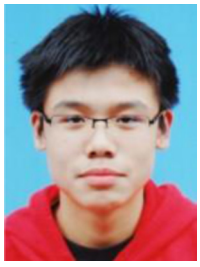
[1] Z.-J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, Visual query suggestion: towards capturing user intent in internet image search, ACM Trans. Multimedia Comput. Commun. Appl. 6 (3) (2010) 13–39.
[2] Z.-J. Zha, M. Wang, Y.-T. Zheng, Y. Yang, Interactive video indexing with statistical active learning, IEEE Trans. Multimedia 14 (1) (2012) 17–27.
[3] Z.-J. Zha, H. Zhang, M. Wang, H. Luan, Detecting group activities with multi-camera context, IEEE Trans. Circuits Syst. Video Technol. 23 (5) (2013) 856–869.
[4] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4nd ed., Academic Press, New York, 2009.
[5] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[6] L. Wang, The Support Vector Machines: Theory and Applications, Springer, Berlin, 2005.
[7] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, 2000.
[8] E. Osuna, R. Freund, F. Girosit, Training support vector machines: an application to face detection, in: International Conference on Computer Vision and Pattern Recognition (CVPR), 1997, pp. 130–136.
[9] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, 1992, pp. 144–152.
[10] L. Kaufman, Solving the quadratic programming problem arising in support vector classification, in: B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999, pp. 147–168.
[11] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999, pp. 185–208.
[12] D. Pavlov, J. Mao, B. Dom, Scaling-up support vector machine using boosting algorithm, in: International Conference on Computer Vision and Pattern Recognition (CVPR), 2000, pp. 219–222.
[13] Y.J. Lee, O.L. Mangasarian, RSVM: reduced support vector machines, in: First SIAM International Conference on Data Mining, 2001.
[14] C.C. Chang, Y.J. Lee, Generating the Reduced Set by Systematic Sampling, Lecture Notes in Computer Science, vol. 3177, Springer, 2004, pp. 720–725.
[15] L.J. Chien, C.C. Chang, Y.J. Lee, Variant methods of reduced set selection for reduced support vector machines, J. Inf. Sci. Eng. 26 (1) (2010) 183–196.
[16] G. Schohn, D. Cohn, Less is more: active learning with support vector machines, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 839–846.
[17] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (2001) 45–66.

[18] M. Li, I. Sethi, Confidence-based active learning, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1251–1261.
[19] I.W. Tsang, J.T. Kwok, P.M. Cheung, Core vector machines: fast svm training on very large data sets, J. Mach. Res. 6 (2005) 363–392.
[20] Q. Gu, J. Han, Clustered support vector machines, J. Mach. Learn. Res. 31 (2013) 307–315.
[21] H. Yu, J. Yang, J. Han, X. Li, Making svms scalable to large data sets using hierarchical cluster indexing, Data Min. Knowl. Discov. 11 (3) (2005) 295–321.
[22] M. Awad, L. Khan, F. Bastani, I. Yen, An effective support vector machine (svms) performance using hierarchical clustering, in: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 663–667.
[23] B. Daniael, D. Cao, Training support vector machine using adaptive clustering, in: Proceedings of SIAM International Conference on Data Mining, 2004.
[24] J. Cervantes, X. Li, W. Yu, Support vector machine classification based on fuzzy clustering for large data sets, in: MICAI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 4293, Springer, 2006, pp. 572–582.
[25] J. Cervantes, X. Li, W. Yu, K. Li, Support vector machine classification for large data sets via minimum enclosing ball clustering, Neurocomputing 71 (2008) 611–619.
[26] L.R. Jen, Y.J. Lee, Clustering model selection for reduced support vector machines, in: Z.R. Yang, H. Yin, R.M. Everson (Ed.), IDEAL 2004, Lecture Notes in Computer Science, vol. 3177, Springer, 2004, pp. 714–719.
[27] S.W. Purnami, J.M. Zain, T. Heriawan, An alternative algorithm for classification large categorical dataset: k-mode clustering reduced support vector machine, Int. J. Database Theory Appl. 4 (1) (2011) 19–29.
[28] H.P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, V. Vapnik, Advances in Neural Information Processing Systems, in: L.K. Saul, Y. Weiss, L. Bottou (Eds.), Parallel Support Vector Machines: The Cascade SVM, 17, MIT Press, 2005, pp. 521–528.
[29] C. Yin, Y. Zhu, S. Mu, S. Tian, Local support vector machine based on cooperative clustering for very large-scale dataset, in: Proceedings of International Conference on Natural Computation, 2012, pp. 88–92.
[30] E.Y. Chang, K. Zhu, H. Wang, J. Li, Z. Qiu, H. Cui, Parallelizing support vector machines on distributed computers, in: Proceedings of NIPS, 2007.
[31] A. Navia-Vázquez, D. Gutiérrez-González, E. Parrado-Hernández, J.J. Navarro-Abellan, Distributed support vector machines, IEEE Trans. Neural Netw. 17 (4) (2006) 1091–1097.
[32] Y. Lu, V. Roychowdhury, L. Vandenberghe, Distributed parallel support machines in strongly connected networks, IEEE Trans. Neural Netw. 19 (7) (2008) 1167–1178.
[33] P.A. Forero, A. Cano, G.B. Giannakis, Consensus-based distributed support vector machines, J. Mach. Learn. Res. 11 (2010) 1663–1707.
[34] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27, software available at ⟨http://www.csie.ntu.edu.tw/cjlin/libsvm⟩.
[35] M. Inaba, N. Katoh, H. Imai, Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering, in: Proceedings of 10th ACM Symposium on Computational Geometry, 1994.
[36] A. Asuncion, D. Newman, UCI machine learning repository ⟨http://archive.ics.uci.edu/ml/⟩, 2007.
[37] Visual Object Classes Challenge 2012 ⟨http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/⟩, 2012.
[38] A. Vedaldi, B. Fulkerson, Vlfeat—an open and portable library of computer vision algorithms, in: Proceedings of the 18th Annual ACM International Conference on Multimedia, 2010.
[39] Li. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: CVPR, 2005, pp. 524–531.

Xiang-Jun Shen received the M.S. degree in Computer Science in 2003 from the Jiangsu University, China, and the Ph.D. degree in Automation in 2006 from the University of Science and Technology, China. He is currently an Associate Professor with the Department of Software Engineering, School of Computer Science and Communication Engineering, Jiangsu University. His research interests in the areas of pattern recognition, peer-to-peer computing, distributed multimedia communication and informational retrieval. He is a member of the ACM and the CCF.

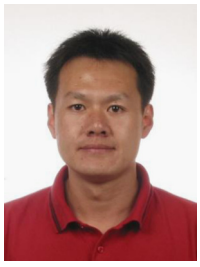
Lei Mu received his B.S. degree in Applied Mathematics from Shaanxi University of Technology in 2011 and is currently a master of JiangSu University. His research interests are in the areas of the image classification and object detection.

Zhen Li received her B.S. degree in Network Engineering from Zhoukou Normal University in 2011 and is currently a master of JiangSu University. Her research interests are in the areas of machine learning with emphasis on distributed clustering and classification, simulation and performance analysis.



Hao-Xiang Wu received his B.S. degree in Software Engineering from Jiangsu University in 2013. His research interests are in the field of data mining, simulation and performance analysis.

Jian-Ping Gou was born in Sichuan, China. He received the B.S. degree in Computer Science from Beifang University of Nationalities, Yinchuan, China, in 2005, the M.S. degree in Computer Science from the Southwest Jiaotong University, Chengdu, China, in 2008, and the Ph.D. degree in Computer Science from University of Electronic Science and Technology of China, Chengdu, China, in 2012. He is currently a lecturer in School of Computer Science and Telecommunication Engineering, JiangSu University, ZhenJiang, JiangSu, China. His current research interests include pattern classification, machine learning. Now he is a IEEE student member.



Xin Chen received the Ph.D. degree in Mechanical Engineering from University of Hawaii at Manoa, Honolulu, HI, USA, in December 2007, and Master degree in Automation from Hefei University of Technology, Hefei, Anhui, China, in July 2003. Currently, he is a Senior Software Engineer at Hermes Microvision Inc., San Jose, CA, USA. From 2011 to 2012, He was an Imaging Scientist at Konica Minolta Laboratory U.S.A., INC. Prior to it, he held an Assistant Computer Scientist at Massachusetts General Hospital (MGH) and Instructor (Research Faculty) at Harvard Medical School (HMS) from 2010 to 2011. He was a Research Associate from 2008 to 2009 at Department of Radiology and Biomedical Imaging, University of California, San Francisco. His primary interests include image processing, pattern recognition, real-time and distributed systems.