K-SVM: An Effective SVM Algorithm Based on K-means Clustering

Yukai Yao, Yang Liu, Yongqing Yu, Hong Xu, Weiming Lv, Zhao Li, Xiaoyun Chen*
School of Information Science and Engineering, Lanzhou University, Lanzhou, China, 730000
Email: yaoyukai@163.com
liuyang11lzu@163.com
yqyu.cn@gmail.com
{hxu11, lvwm10, lizh10, chenxy}@lzu.edu.cn

***Abstract*—Support Vector Machine (SVM) is one of the most popular and effective classification algorithms and has attracted much attention in recent years. As an important large margin classifier, SVM dedicates to find the optimal separating hyperplane between two classes, thus can give outstanding generalization ability for it. In order to find the optimal hyperplane, we commonly take most of the labeled records as our training set. However, the separating hyperplane is only determined by a few crucial samples (Support Vectors, SVs), we needn't train SVM model on the whole training set. This paper presents a novel approach based on clustering algorithm, in which only a small subset was selected from the original training set to act as our final training set. Our algorithm works to select the most informative samples using K-means clustering algorithm, and the SVM classifier is built through training on those selected samples. Experiments show that our approach greatly reduces the scale of training set, thus effectively saves the training and predicting time of SVM, and at the same time guarantees the generalization performance.**

***Index Terms*—SVM model, K-means clustering, Kernel function, predict**

I. INTRODUCTION

With the rapid development of pattern recognition and machine learning, a lot of data mining algorithms have been proposed by experts, through which researchers can find much interesting hidden information from the observational data, classification information is one of the most important ones in it, which can be used to recognize, predict or classify those current unseen data. Generally, machine learning algorithms can be divided into three categories: Supervised Learning algorithm, Unsupervised Learning algorithm, and Semi-supervised Learning algorithm. The common supervised learning algorithms include regression analysis and classification analysis. Data classification process includes two stages: the first one is the learning stage, the aim of which is to build a classifier through analyzing the labeled data; the second one is the predicting stage, which using the established model for predicting. The model should have enough generalization ability, i.e., that the model not only has good classification performance on the training data, but also has a high classification accuracy for the future data, which supposed has the same statistical distribution as the training data. The main classification algorithms include Decision Tree, Bayes, Neural network, Support Vector Machine (SVM), etc.

Support Vector Machine (SVM) [1-2] is one of the most popular and effective algorithms in machine learning. SVM is based on the structural risk minimization criterion and its goal is to find the optimal separating hyperplane where the separating margin should be maximized. This approach improves the generalization ability of the learning machine and solves some problems like non-linear, high dimension data separation and the classification issue that lacking of prior knowledge. SVM is used in systems for face recognition [3-4], road sign recognition and other similar application areas [5] because of its sound theoretical foundation and good generalization ability in practical application.

SVM works well in both linear and non-linear conditions, and finding the optimal separating hyperplane is the key to separate data. For non-linear situations, SVM exploits the kernel trick to map low-dimension data into high dimension feature space. In the practical application, SVM makes use of all the labeled data to find the separating rule, but training on large scale data can bring with higher computation cost. In order to decrease the computational complexity, the solution that can be exploited includs two species, one is to improve the algorithm itself, such as the Least Square SVM [6-7], the SMO [8] (Sequential Minimal Optimization) under semi-positive definite kernel; the other is to decrease the number of input vectors.

The main task of clustering [9] is to group the objects into clusters, objects in the same cluster are more semblable than those in different clusters. It can find the relationships among data objects in an unsupervised way. A lot of clustering algorithms have been proposed and improved, aiming to enhance the efficiency and accuracy. According to cluster mode, clustering algorithms can be categorized into: centroid-based clustering, hierarchical clustering, distribution-based clustering and density-based clustering, etc.

In this paper, we present an efficient approach for minimizing the sample space. The final training data are

doi:10.4304/jcp.8.10.2632-2639

selected from the clustering result; they have massive information and important contribution for building SVM model, and can effectively decrease the scale of training set on the premise of guaranteeing the accuracy of the model.

The rest of this paper is organized as follows: section II is a brief introduction of SVM and K-means clustering. The details of the improved algorithm are described in section III. Section IV is mainly about the experiments. Section V is the conclusion of this paper.

II. Related Work

A. Support Vector Machine

In cases of linear separable and binary classification, the goal of SVM is to find an optimal hyperplane [10] which can separate the two classes obviously with a maximal separating margin. The margin is defined as the geometrical distance of blank space between the two species. The greater of the margin, the better of the generalization ability of SVM classifier. The boundaries of this margin are hyperplanes paralleling with the separating hyperplane.

Supposed a training set $X=\{x_1,x_2,...,x_n\}$ and the corresponding label set $Y=\{y_1,y_2,...,y_n\}$, a sample can be expressed as the expression below: $\{(x_i,y_i)\}$, $x_i \in R^d$, $y_i \in \{+1,-1\}$, $i \in \{1,2,...,n\}$, where d is dimension number of the input space, n is the number of samples. For standard SVM, w and b are the weight vector and bias of the optimal hyperplane, respectively. The separating function can be written as formula 1.

$$wx+b=0 \tag{1}$$

We can modulate formula 1, so as to let samples in the dataset meet the following requirements in formula 2 and formula 3:

$$wx_i+b \geq +1, \text{ where } y_i=+1. \tag{2}$$

$$wx_i+b \leq -1, \text{ where } y_i=-1. \tag{3}$$

and the boundary function of the separating margin can be defined with formula 4:

$$\begin{aligned} wx+b&=1, \\ wx+b&=-1. \end{aligned} \tag{4}$$

For all the data points in the training set, they must satisfy the following constraints (Linear Separable Problem):

$$y_i(wx_i+b) \geq 1 \tag{5}$$

According to the definition of separating margin and the formulas of the separating hyperplane and separating boundaries, it is not hard to get the value of the classifying margin, which is $2/\|w\|$. Maximizing the value of the separating margin is equal to minimizing the value of $\|w\|^2$. Generally, we solve this constrained optimization problem with Lagrange multipliers. We construct the following Lagrange function with formula 6:

$$L(w,b,\alpha)=\frac{1}{2}w^Tw-\sum_{i=1}^{n}\alpha_i[y_i(wx_i+b)-1] \tag{6}$$

Easily, we can obtain the following formula 7:

$$\begin{aligned} w&=\sum_{i=1}^{n}\alpha_i y_i x_i \\ \sum_{i=1}^{n}\alpha_i y_i&=0 \end{aligned} \tag{7}$$

, and we substitute the formula 7 into the Lagrange function 6 and then get the corresponding dual problem, which is described as formula 8:

$$\begin{aligned} \max_{\alpha} W(\alpha)&=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ s.t.\sum_{i=1}^{n}\alpha_i y_i&=0 \\ \alpha_i &\geq 0, i=1,...,n \end{aligned} \tag{8}$$

And the Karush-Kuhn-Tucker complementary conditions are formula 9:

$$\begin{aligned} \alpha_i[y_i(wx_i+b)-1]&=0, \\ i&=1,...,n \end{aligned} \tag{9}$$

Consequently, we can distinguish the Support Vectors (SVs) from the other vectors. These vectors whose α_is are nonzero can be called SVs, they are used to determine the optimal separating hyperplane.

The dual problem (the original problem are called Primal Problem.) in formula 8 is a typical convex quadratic programming optimization problem. After determined the Lagrange multipliers α_i^*, we can get the w through equation 7.

$$w^*=\sum_{i=1}^{n}\alpha_i^* y_i x_i \tag{10}$$

According to the information of SVs, we also can compute the b^*, then the optimization hyperplane function can be obtained.

$$b^*=1-w^{*T}x_{SV} \quad \text{for} \quad y_{SV}=+1 \tag{11}$$

But in practical applications, we could not find a clear separating hyperplane to differentiate the data, because of the complexity of dataset. In such conditions, we allow a few samples existing in the wrong side of the separating hyperplane, and accordingly, the maximal margin classifier in this pattern is so called Soft Margin SVM [11-12]. The constraints become:

$$y_i(wx_i+b) \geq 1-\xi_i \tag{12}$$

where ξ_i is called slack variable. The idea of "Soft Margin" aims to improve the generalization ability of SVM.

In order to maximize the margin, the optimization problem is equivalent to a quadratic programming problem [12], which can be expressed with formula 13:

$$\min_{w,b} \frac{1}{2}||\mathrm{w}||^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad \mathrm{y_i}(\mathrm{wx_i}+\mathrm{b}) \geq 1\text{-}\xi_\mathrm{i}, \xi_i \geq 0, i = 1,2,...,n \quad (13)$$

where C is called penalty parameter, and it is a positive real constant, which is selected by user and is used to keeps the tradeoff between the complexity and the number of misclassified data. Using the Lagrange multipliers, we can transform the original optimizing problem into its dual problem, which is described in formula 14.

$$\max_{\alpha} L(\mathrm{w},\mathrm{b},\alpha) = \sum_{i=1}^{n}\alpha_i \text{-} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$
$$s.t. \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{y_i}\alpha_i = 0, \alpha_i \geq 0, i = 1,2,...,n \quad (14)$$

where α_i is the Lagrange multiplier with respect to the i th inequity. And at the same time, the Karush-Kuhn-Tucker complementary conditions are [13]:

$$\alpha_i\left[\mathrm{y_i}(\mathrm{wx_i}+\mathrm{b})\text{-}1+\xi_\mathrm{i}\right] = 0, \quad \alpha_\mathrm{i}\xi_i = 0, i = 1,2,...,n \quad (15)$$

Therefore, the examples are on or close to the boundary hyperplanes of maximal margin are called Support Vectors (SVs), those vectors whose α_i s are nonzero determine the optimal separating hyperplane [10], and all the other α_i s are equal to zero.

Finally, we can get the decision function which is decided by the Lagrange multipliers and SVs, and is described in formula 16.

$$f(\mathrm{x}) = \mathrm{sgn}(\sum_{\mathrm{i}=1}^{\mathrm{n}}\alpha_\mathrm{i} y_i(\mathrm{x_i x})+\mathrm{b}) \quad (16)$$

In order to solve the non-linear issues, Kernel technique [14-16] is taken to map the low-dimension data into high dimension even infinite dimension feature space, through which the non-linear separable problem can be transformed into linear separable issue. Kernel technique can effectively avoid the emergence of non-linear mapping $\varphi: R^d \to H$, which means sample x_i in the original space is mapped into $\varphi(x_i)$ in the high dimension feature space H . Using the Kernel function, inner product in the mapped feature space can be replaced with Kernel function: $K(\mathrm{x_i},\mathrm{x_j}) = \varphi(\mathrm{x_i})\cdot\varphi(\mathrm{x_j})$, the essence is to substitute the inner products in the feature space with inner products in the original data space. The final decision function (formula 16) is revised to the form described in formula 17:

$$f(\mathrm{x}) = \mathrm{sgn}(\sum_{\mathrm{i}=1}^{\mathrm{n}}\alpha_\mathrm{i}\mathrm{y_i}\mathrm{K}(\mathrm{x_i},\mathrm{x})+\mathrm{b}) \quad (17)$$

B. K-means

Clustering is aims to divide the data into groups. And each group is constructed by similar data, in other words, it means that the similarity between dates in the same group is smaller than others.

K-means is a clustering algorithm in data mining field. It is used to cluster analysis, and has a high efficiency on data partition especially in large dataset. As an unsupervised learning algorithm, we do not know the result clusters before executing the algorithm, it is unlike classification. Because the number of the cluster is unknown, so it usually takes the desired number of groups as input, and in the real applications, we commonly decide it use the experience value [17-19].

K-means is a very simple algorithm based on similarity. The measure of similarity plays an important role in the process of clustering. The similar points are assigned to a same cluster, and the dissimilar ones in different cluster. We usually use Euclidean Distance to measure the similarity between two data points.

The different metric method of similarity will not change the result, but the result of K-means is more sensitive to the initial centroids. The two factors are: the one is the value of K, and another is the initial selection of centroids. It implements the iterative technique. This process will not stop until the mean value of all the clusters not change. That is to say, the grouping is done by minimizing the sum of squared distances between the objects and the corresponding centroids [17-19].

In K-means algorithm, the choosing of initial center is the key to get precise result. If choosing the proper initial centroids will get a good result, but if it is not, the result will get worse, it may make a large and low density cluster divided into pieces, or merge two close clusters into a one group, etc. So we usually choose the initial centroids randomly, or use the priori knowledge to label some of them to achieve a good result.

The K-means will iterates between two steps until converge. The first step is assign each point to the closet cluster based Eucliean distance. The second step is updating the mean value of cluster or the cluster centroid. Note that each iteration needs $N \times K$ comparison, N is the number of the dataset, and K is the number of clusters that we desired to get.

K-means [17-19] needs only one parameter that denoting the number of clusters. This illustrates that it is simple and effectively. It is a most common algorithm using an iterative refinement technique. Given a dataset of n-dimensional vectors, $D = \{\mathrm{x_i}|\mathrm{i} = 1,2,...,\mathrm{N}\}$ where $x_i \in R^n$ is the i th data point. Each point falls in one and only one partition, and it belongs to the nearest cluster. The algorithm is deemed to have converged when the state of assignments process keeps no change. As it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may

depend on the initial clusters. As the result of K-means algorithm is uncertain, we usually run it multiple times, and cluster result is determined through vote mechanism.

The steps of this algorithm are described as follows.

Steps of K-means:

1. At the beginning we randomly choose K points in the database as the initial cluster center.
2. **repeat**
3. each object is assigned to the most similar cluster based on the mean value of the objects in a cluster.
4. update the mean value of a cluster
5. **until** the mean values of clusters not change.

III. K-SVM: SVM ALGORITHM BASED ON K-MEANS CLUSTERING

A. *The Rule of Selecting Samples*

The goal of our algorithm is to sift the important samples, that is to say, to find the most important samples with massive information. In Section II, we have already mentioned the definition of Support Vectors (SVs), which are commonly the boundary points between clusters. However, the boundary points are easy to be misclassified in clustering. Inspired by this problem, our algorithm uses the clustering approach to select boundary points[20] which are more likely to be SVs.

Cluster assumption [21-22] is our theoretical basis of selecting the most informative samples, it can be described as: if the distance between two samples is relatively close, then the samples are apt to have the same classification label. In other words, the intra-cluster distance is smaller than inter-cluster distance. So the decision hyperplane between clusters should be located at the sparse region. In the premise of Cluster assumption, the learning algorithm could analyze the data distribution in feature space and tune the location of the decision boundary.

In order to find the decision boundary, we can search the points on or close to the boundary of clusters. The locations of these points determine the decision boundary. The points are difficult to be labeled, if the distance of one of them to the opposite cluster is close enough, and it is more easily to be misclassified in clustering process. For the binary classification problem in Figure 1., the blue lines are the boundary hyperplanes of the maximal margin which we are going to find. They are determined by the six black points and they are located near or on the boundaries of two clusters, respectively. All the points in Figure 1 form the original training set; the goal of our algorithm is to select a small set which contains the six black points. So our approach aims to use the clustering algorithm to select the misclassified points and according to the difference of the labels of these samples and their neighbors to select some of them for training. Thus we can reduce the scale of the training set effectively.

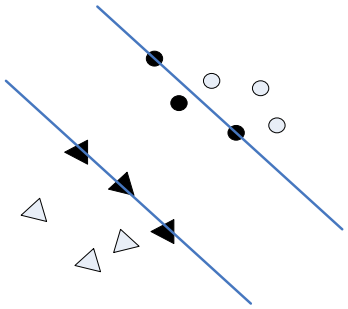


Figure 1. Sketch map of binary classification

The blue lines are the boundary hyperplanes, the margin is maximized, and the black points are the SVs.

The black points in Figure 1. is the misclassified ones in the clustering process, we can use the information of them for training and building the prediction model. Suppose that we only consider the position of the misclassified points, it has several situations is shown in Figure 2..

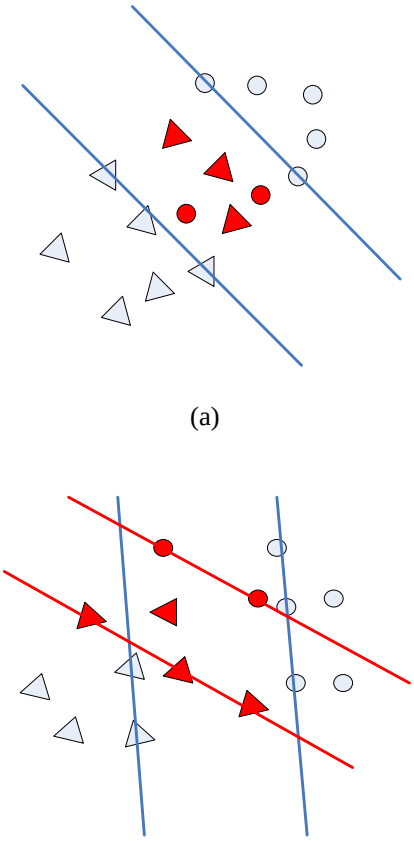


Figure 2. the position of red misclassified points

Figure 2. is used to show the two situations that may occur. For the Figure 2. (a), it can be seen that the red misclassified points are mixed. If we use the information offered by them, the accuracy of the training model is

worse. In order to solve the problem of this situation, these points can be removed from the original dataset, and use the rest to train the SVM model. But this approach is not efficient because the number of the misclassified points is few sometimes; training the rest data has a little significance.

And for Figure 2. (b), the blue lines are the proper hyperplines, the red ones are the hyperlines built by red misclassified points. We can see that there has a great deviation compared with the original hyperplanes.

In order to solve those problems, we start it from the neighbors of misclassified points. In Figure 3. and Figure 4., the red points are the misclassified ones, and the black ones are their neighbors. We want to choose the neighbors whose true labels are different to the misclassified points that are shown in Figure 3., or the number of the points with the same label is equal to that with the different label which is shown in Figure 4.. We believe that those points have massive information.

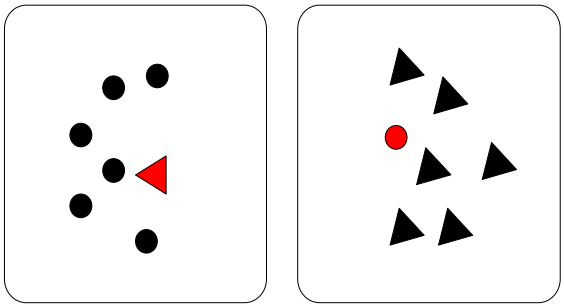


Figure 3. the black points are which to be selected

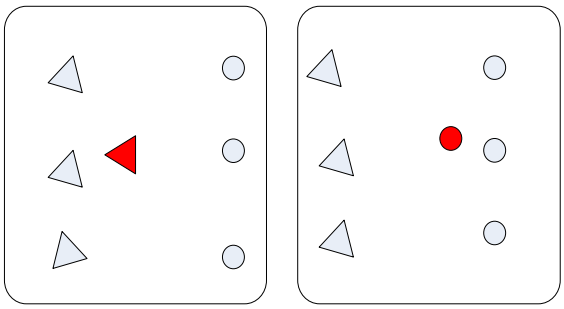
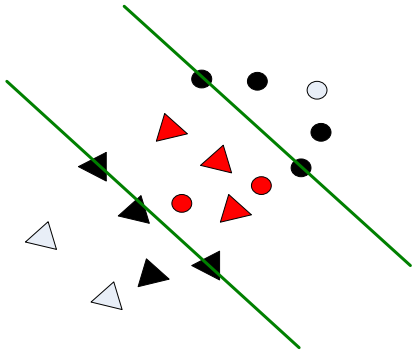


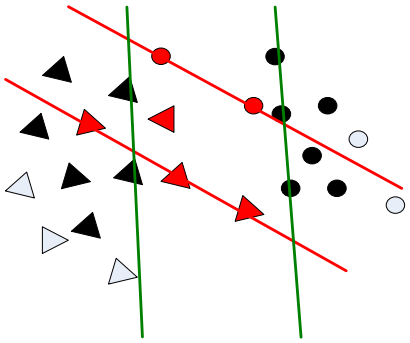
Figure 4. the red misclassified points are the most uncertain ones that are selected.

According to the analysis above, the implementation process of our algorithm can be divided into two steps. In the first step, we run the cluster algorithm on the original training set; as a result, each data point will get a cluster label. Then we will compare the cluster label of a data with its true label, if they are different, we have reason to doubt that they may locate in the edge region of the two clusters. In the second step, we check the labels of its N-nearest neighbors; choose some of them as the candidate training set. The way to choose points is shown in Figure 5:



(a)

In Figure 5, the red points represent those misclassified points and we will choose the black ones in the Figure 5. (a) and (b) to construct the candidate training set.



(b)

Figure 5. The way of choosing point

As can be seen from Figure 5, it is obvious that the way of choosing samples is reasonable. In Figure 5, the green lines are the adjusted boundary hyperplanes, and the red lines in Figure 5. (b) are the worse ones. The experiments show that the old ones and the new ones almost have the same generalization ability.

B. Algorithm Description

Given a training set $X=\{x_1,x_2,...,x_n\}$ and the corresponding label set $Y=\{y_1,y_2,...,y_n\}$, a sample can be expressed as $\{(x_i,y_i)\}$. At the beginning we use K-means on training set X, each sample will get a cluster label, if the cluster label of a data is different with its true label, and then it will be added into the misclassified set $misC$. Let y_{ik} be the cluster label of x_i after the clustering process and y_i be its true label, we define the misclassified set with formula 10:

$$misC=\{x_i|y_{ik}\neq y_i\} \quad (10)$$

For each member in this set, we find its N-nearest neighbors. We define $NNeighbor$ as a sample's N-nearest neighbor set and $NNLabel$ as its corresponding label set.

Algorithm I :K-SVM
Input: X, Y
Output: Model
begin
1. $misC = \varnothing$, $NT = \varnothing$ /*initialize*/
2. K-means(k, X) /* k is given by the user or by prior knowledge*/
3. for each $x_i \in X$ {
4. if $y_{ik} \neq y_i$
5. $misC = misC \cup x_i$;}
6. for each $x_j \in misC$ {
7. $NNeighbor_{jm} = findNeighbor(\mathrm{x}_j, \mathrm{n})$
/* n is the number of neighbors and $NNeighbor_{jm}$ represents the m th neighbor of the j th sample.*/
8. $NNLabel_{jm}$ /* $NNLabel_{jm}$ is the corresponding label set of $NNeighbor_{jm}$ */
9. if $|NNlabel_{jm} \neq y_i| = |NNlabel_{jm} = y_i|$
10. $NT = NT \cup \{\mathrm{x}_j\}$
11. if $|NNLabel_{jm} \neq y_i|$
12. $NT = NT \cup NNeighbor_{jm}$ }
13. $\mathrm{model} = svmtrain(NT, NTlabel)$
end

According to the way of selecting samples, we pick out those points that meet requirements and add them into the new training set NT. After sifting, we will get a small subset comparing with the original training set, which are utilized to train and predict. Algorithm I describes the details of our algorithm with pseudo code.

Algorithm II**: findNeighbor(x,n)**
Input: X, x, n
Output: $NNeighbor$
begin
1. for each $x_i \in X$ {
2. for each $x_h \in X$ {
3. $if\ x_i \neq x_h,\ dis = \sqrt{\sum_{j=1}^{d} (x_{ij} - x_{hj})^2}$ /* $x \in R^d$, x_{ij} is the j th demension of i th sample*/}}
4. $NNeighbor$ is the set of the samples corresponding to the n-smallest dis values.
end

The way of selecting samples is effective, it can not only reduce the scale of the training set but also guarantee the accuracy and avoid overfitting. Algorithm II aims to find the N-nearest neighbors of a sample.

Our algorithm aims to find the samples which could be SVs as much as possible. This way can save time for establishing SVM model and make the process of training converge as soon as possible. The complexity of calculation depends on the number of SVs, the fewer of SVs, the lower of the computation complexity.

IV. EXPERIMENTS

In our experiments, we put our efforts on solving the binary classification problem, and the training dataset is collected from the original dataset through stratified sampling [23]. Our algorithm and LIBSVM are implemented on three datasets, respectively. Each dataset is divided into two parts, the training set and the testing set. And the experiment results show that the number of SVs decrease greatly and the predict accuracy is the same or a litter higher than that of LIBSVM. The algorithm is implemented with Matlab and libsvm toolbox. The kernel function we used here is the Gaussian kernel : $K(x_i, x_j) = \exp^{-\|x_i - x_j\|^2 / 2\sigma^2}$.

A. Experiment Datasets

TABLE 1:
INFORMATION OF DATASETS

DataSet	Train (Number × dimension)	Test (Number × dimension)
breastcancer	448×9	235×9
heart_scale	180×13	90×13
liverdisorders	230×6	115×6

Table 1 describes the details of datasets in our experiment. These datasets are of binary classification problem which can be downloaded from the Repository of machine learning databases of the well-known University of California at Irvine (UCI) [24]. They are medical data which obtained from real life.

The instances are described by attributes, some of which are linear and some are nominal.

B. Experiment Results

We commonly use accuracy rate [25] to evaluate the performance of the binary classification algorithms. In our experiments, we compare the results from the following aspects: the number of samples in training set, the number of SVs, the accuracy rate, and the approximately consuming time.

The result in Table 2, the new number of training set is a small set which is selected from the original set. It

illustrates that the way of reducing the scale of the training set is effective, the number of the training set decreases in different levels.

TABLE 2:
THE NUMBER OF TRAINING SET

DataSet	The original number of training set	The new number of training set
breastcancer	448	33
heart_scale	180	38
liverdisorders	230	34

TABLE 3:
THE NUMBER OF SVS

DataSet	The original number of SVs	The new number of SVs
breastcancer	69	**14**
heart_scale	96	**26**
liverdisorders	210	**18**

As can be seen in Table 3, the number of SVs obtained from our algorithm reduces obviously, and the performance of our algorithm is almost the same as LIBSVM, because our algorithm excludes the redundant samples which have no contribution for establishing SVM model. Table 4 shows the accuracy rate of our algorithm and LIBSVM.

TABLE 4:
THE ACCURACY RATE

DataSet	The original accuracy rate	The new accuracy rate
breastcancer	98.7234%	99.5745%
heart_scale	84.4444%	84.4444%
liverdisorders	63.4783%	63.4783%

The accuracy rate of our algorithm is almost the same as the original one, in other words, our algorithm can guarantee the generalization ability, and it utilizes fewer SVs but gets the same or better accuracy rate. The approximately consuming time is saved effectively; it is an average value of time after running the algorithms for 5 times. And Table 5 illustrates the comparison of the average time.

The number of training samples determines the computation complexity, if we use the most valuable samples for training, the consuming time can be saved significantly. The consuming time includes the time for establishing SVM model and the time for predicting.

TABLE 5:
THE COMPARISON OF THE AVERAGE TIME

DataSet	The average original time (/S)	The average new time (/S)
breastcancer	0.01273	**0.00720**
heart_scale	0.01263	**0.00519**
liverdisorders	0.01745	**0.00487**

In a word, our way to deal with the training set is available. Our algorithm can not only reduce the scale of training set thus greatly reduce the algorithm complexity, but also guarantee the generalization performance of our algorithm.

V. CONCLUSION

In this paper, we dedicate to solve the problem of reducing the scale of the training set with the method of clustering. We use the K-means clustering approach to select the few most informative samples, which are used to construct our real training set. Experiment results show that our algorithm reaches the goal of reducing the scale of training set, and greatly reduces the training and predicting time, meanwhile assures the generalization ability of our K-SVM algorithm.

In the future, we will try to use our approach on large-scale data set, and explore a more precise way to select SVs.

ACKNOWLEDGMENTS

This paper is supported by the Fundamental Research Funds for the Central Universities (lzujbky-2013-229) and is funded in pary by the Mountain Torrent Disaster Prevention projects from Hydrology and Water Resources Survey Bureau in Qinghai Province. Any opinions, findings, and conclusions or recommend-ations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] V.Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[2] Jing Bai, Lihong Yang, Xueying Zhang, "Paramrter Optimization and Application of Support Vector Machine Based in Parallel Artificial Fish Swarm Algorithm", Journal of Software, pp. 673-679, vol. 8, no. 3, 2013.
[3] Riadh Ksantini, Boubakeur Seddik Boufama, Imran Shafiq Ahmad, "A New KSVM + KFD Model for Improved Classification and Face Recognition", Journal of Multimedia, pp. 39-47, vol. 6, no. 1, 2011.
[4] Guodong Guo, Stan Z.Li, "Face Recognition by Support Vector Machines", Automatic Face and Gesture Recognition, pp.196-201, 2000.
[5] S.Maldonado-Bascon, S.Lafuente-Arroyo, "Road-Sign Detection and Recognition Based on Support Vector Machines", IEEE Transactions Intelligent Transportation Systems, vol. 8, no. 2, pp. 264-278, 2007.
[6] Suykens JAK, Vandewalle J, "Least Squares Support Vector Machine Classifiers", Neural Processing Letters, vol. 9, no.3, pp.293-300, 1999.
[7] L Lukas, JAK Suykens, "Least Squares Support Vector Machines: a multi two-spiral benchmark problem", Procedding of the Intdonesian Student Scientific Meeting, pp.289-292, 2001.
[8] John C.Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization", 1998.
[9] Yu Zong, Ping Jin, Dongguan Xu, Rong Pan, "A Clustering Algorithm based on Local Accumulative Knowledge", Journal of Computers, pp.365-371, vol.8, no.2, 2013.
[10] Corinna Cortes, Vladimir Vapnik, "Support-vector networks", Machine Learning, vol. 20, no. 3, pp.273-297, 1995.
[11] William S Noble, "What is a support vector machine", Nature Biotechnology 24, pp.1565-1567, 2006.
[12] Qiang Wu, "SVM Soft Margin Classifiers: Linear Programming versus Quadratic Programming", Neural Computation, vol. 17, no. 5, pp.1160-1187, 2005.
[13] Guang-Bin Huang, Xiaojian Ding, "Optimization method based extreme learning machine for classification", Neurocomputing, vol. 74, no. 1-3, pp.155-163, 2010.
[14] Colin Campbell, "Kernel methods: a survey of current techniques", Neuroxomputing, vol. 48, no. 1-4, pp.63-84, 2002.
[15] G.Baudat, "Generalized Discrininant Analysis Using a Kernel Approach", Neural Computation, vol. 12, no. 10, pp.2385-2404, 2006.
[16] B. Scholkopf and A.J. Smola, Learning With Kernels: Support Vector Machines, Regularization, Optomization, and Beyond, MIT Press, Cambridge, 2002.
[17] Juanying Xie, Shuai Jiang, Weixin Xie, Xinbo Gao, "An Efficient Global K-means Clustering Algorithm", Journal of Computers,, pp.271-279, vol.6, no.2, 2011.
[18] S.Sujatha, A.Shanthi Sona, "New Fast K-Means Clustering Algorithm using Modified Centroid Selection Method", International Journal of Engineering Research & Technology (IJERT), pp.1-9, vol.2, no.2, 2013.
[19] Lin Yujun, Luo Ting Yao Sheng, Mo Kaikai, Xu Tingting, "An improved clustering method based on k-means", Fuzzy Systems and Knowledge Discovery(FSKD), pp. 734-737, 2012.
[20] Y.T. Gu, G.R. Liu, "A boundary point interpolation method for stress analysis of solids", Computational Mechanics, vol. 28, no. 1, pp. 47-54, 2002.
[21] Olivier Chapelle, Jason Weston, "Cluster Kernels for Semi-Supervised Learning", Advances in Neural Information Processing Systems 15, pp. 585-592, 2003.
[22] Olivier Chapelle, Alexander Zien, "Semi-Supervised Classification by Low Density Separation", 10th Int'l Workshop Artificial Intelligence and Statistics, pp. 57-64, 2005.
[23] Freedman David, Pisani Robert, Statistics, Norton, New York, 2007.
[24] C. Blake, C. Merz, "UCI Repository of Machine Learning Databases",Available:http://www.ics.uci.edu/~mlearn/ML Repository.html, 1998 [Online].
[25] P. Williams, S. Li, "A geometrical method to improve performance of support vector machine", IEEE Transactions on Neural Network, vol. 18, no. 3, pp. 942-947, 2007.

Yukai Yao received the BS degree from the Department of Computer Science and Technology, Northwest Normal University, in 1997, and the Ma degree from School of Information Science and Engineering, Lanzhou University, in 2011. He is now a doctoral student in the School of Information Science and Engineering, Lanzhou University. His research interests include high performance computing, pattern recognition and data mining. He is a member of CCF and IET.

Yang Liu received the BS degree from the School of Computer Science and Technology, Zhengzhou University, in 2011. She is now a master student of the School of Information Science and Engineering, Lanzhou University. Her research interests include Artificial Intelligence, Data Warehouse and Data Mining. She is a member of IET.

Xiaoyun Chen received the BS degree from the Department of Computer Technology, Jilin University, and the MA degree from the Institute of Atomic Energy, Chinese Academy of Sciences, in 1995. Professor, PhD supervisor, the Director of Institute of Computer Software and Theory, the Director of national Linux Technical Training and Promotion Center of Lanzhou University, the Director of IBM Technology Center of Lanzhou University, Senior Member of CCF, the member of Database Special Committee, the member of Theoretical Computer Science Special Committee, the member of Liberal Arts Computer Basic Teaching Instruction Committee of the Ministry of Education. The main research fields are data warehouse design and construction, data mining algorithms and applications, special data mining, high performance computing, parallel data mining, search engine technology and method, weather information processing, big data, etc.