Methods of Decreasing the Number of Support Vectors via k -Mean Clustering

Xiao-Lei Xia[1], Michael R. Lyu[2], Tat-Ming Lok[3], and Guang-Bin Huang[4]

[1] Institute of Intelligent Machines, Chinese Academy of Sciences,
P.O.Box 1130, Hefei, Anhui, China
xlxia@iim.ac.cn
[2] Computer Science & Engineering Dept., The Chinese University of Hong Kong,
Shatin, Hong Kong
[3] Information Engineering Dept., The Chinese University of Hong Kong,
Shatin, Hong Kong
[4] School of Electrical and Electronic Engineering, Nanyang Technological University

Abstract. This paper proposes two methods which take advantage of k -mean clustering algorithm to decrease the number of support vectors (SVs) for the training of support vector machine (SVM). The first method uses k -mean clustering to construct a dataset of much smaller size than the original one as the actual input dataset to train SVM. The second method aims at reducing the number of SVs by which the decision function of the SVM classifier is spanned through k -mean clustering. Finally, Experimental results show that this improved algorithm has better performance than the standard Sequential Minimal Optimization (SMO) algorithm.

1 Introduction

Support Vector Machine (SVM) [1] is a new class of approaches for classification and regression problems. Currently, SVMs are gaining popularity due to attractive features and have been successfully applied to various fields. Unlike previous machine learning algorithms such as traditional neural network models [2, 3, 4, 5], the SVM developed by Vapnik is derived from statistical learning theory and employs the structural risk minimization (SRM) principle [1], which can significantly enhance SVM's generalization capability. With a clear geometrical interpretation, the training of the SVM is guaranteed to find the global minimum of the cost function.

In general, training an SVM requires the solution of a very large quadratic programming (QP) optimization problem. The large size of the training sets typically used in applications is a formidable obstacle to a direct use of standard quadratic programming techniques [6]. Recently, many algorithms have been developed to solve the problem [7, 8, 9]. The most typical one is John Platt's Sequential Minimal Optimization (SMO) [10], which breaks a large QP problem into a series of smallest possible QP problems. SMO is generally fast and efficient for linear SVMs and sparse data sets. However, the number of support vectors (SVs) that SMO produces is too large in proportion to the size of the input dataset for training SVM. It is shown that if

D.S. Huang, X.-P. Zhang, G.-B. Huang (Eds.): ICIC 2005, Part I, LNCS 3644, pp. 717 – 726, 2005.

the training vectors are separated without errors the expectation value of the probability of committing an error on a test example is bounded by the ratio between the expectation value of the number of support vectors and the number of training vectors:

$$E[\Pr(error)] \leq \frac{E[number\ \ of\ \ support\ vectors]}{number\ \ of\ training\ vectors}. \tag{1}$$

From inequality (1), it can be drawn that a small number of support vectors can lead to a small testing error and also a SVM with a better generalization capability.

In this paper k -mean clustering [11, 13] provides two methods to suppress the number of support vectors based on SMO algorithm in the training of SVM. For the first method, k -mean clustering method helps pick a set smaller than the original dataset to train SVM, which dramatically reduce the number of SVs without reducing the training correctness. It also can be concluded that with the decrease in the number of training examples the computational time that SMO requires greatly falls. The other application of k -mean clustering aims at finding less support vectors to describe the normal vector of the optimal hyperplane of SVM. The normal vector is spanned by a number of the mapping of SVs from input space into feature space where the kernel trick plays an essential role [6, 12]. k - mean clustering can help find a certain number of support vectors whose feature space image would well approximate the expansion. The two methods can suppress the number of SVs and result in a SVM with significant efficiency and outstanding generalization ability.

The paper is organized as follows. Section 2 gives a brief introduction to the theoretical background with reference to classification principles of SVM. Section 3 describes the two methods for the decrease in the number of SVs. Experimental results is demonstrated in Section 4 to illustrate the efficiency and effectiveness of our algorithm. Conclusions are included in Section 5.

2 Support Vector Machine

Consider the problem of separating the set of N training vectors belonging to two classes, where $x_i \in R^n$ is the i th input data and $y_i \in R$ is the i th output data

$$(x_1, y_1),...,(x_N, y_N) \in R^n \times Y, \qquad Y = \{-1, +1\} \tag{2}$$

with a hyperplane

$$\textbf{H:} \qquad \langle w, x \rangle + b = 0 \tag{3}$$

where w is normal to the hyperplane and $b / \| w \|$ is the perpendicular distance from the hyperplane to the origin. The hyperplane is regarded as optimal if all the training vectors are separated without error and the margin (i.e. the distance from the closest vector to the hyperplane) is maximal. Without loss of generality, it is appropriate to consider a canonical hyperplane, acquired by rescaling w and b so that the vectors $x_i \ (i = 1,..., N)$ closest to the hyperplane satisfy:

$$|\langle w\ ,\ x_i\rangle + b| = 1 . \tag{4}$$

Hence, the margin is $2/\|w\|$. Thus the hyperplane $<w,b>$ is given by the solution to the following optimization problem:

$$\begin{aligned} &\min_{\mathrm{w,\,b}} \quad \frac{1}{2} w^T w \\ &\text{subject to} \quad y_i\left(w^T x_i + b\right) \geq 1 \\ &\forall i = 1, 2, \ldots, N \end{aligned} \tag{5}$$

The training vectors for which the equation (4) holds are termed as support vectors (SV).The equivalent dual problem to equation (5) can be written as:

$$\begin{aligned} &\max\ Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \left\langle x_i\ , x_j \right\rangle \\ &\text{subject to} \quad \sum_{\mathrm{i=1}}^{N} \alpha_i y_i = 0 \\ &\forall i = 1, 2, \ldots, N \end{aligned} \tag{6}$$

where the α_i are the Lagrange multipliers and constrained to be non-negative. The linear SVM classifier can be denoted as:

$$f(x) = \operatorname{sgn}\left(\langle w, x\rangle + b\right) . \tag{7}$$

With respect to the case of nonlinearly separable datasets, SVM employs a kernel function K to implement the dot product between the functions $\phi(x_i)$ which can map the data from input space to a high dimensional feature space H , i.e.:

$$K\left(x_i, x_j\right) = \left\langle \phi\left(x_i\right), \phi\left(x_j\right)\right\rangle . \tag{8}$$

The theory of functional analysis suggests that an inner product in feature space correspond to an equivalent kernel operator in input space provided that K satisfies Mercer's condition.

Furthermore, variables ξ_i $\left(i = 1, 2, \ldots, N\right)$ are introduced to allow the margin constraints to be violated while C determines the tradeoff between error and margin. Then a quadratic optimization problem is introduced as follows:

$$\begin{aligned} &\min \quad \frac{1}{2} w^T w + C\left(\sum_{i=1}^{l} \xi_i^2\right) \\ &\text{subject to} \quad y_i\left(w^T \phi\left(x_i\right) + b\right) \geq 1 - \xi_i \end{aligned} \tag{9}$$

The decision function of the nonlinear classifier is:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b \right) \cdot \tag{10}$$

3 Reducing the Number of Support Vectors via k-Mean Clustering

3.1 The First Method Using k-Mean to Reduce the Number of SVs

The first method used to modify the conventional SMO is to employ the k-mean clustering to choose a set which reflects the general features of the full input dataset but has much fewer data points. The approach is based on the observation that in many cases a large proportion of the original input dataset is redundant for training SVM. A good SVM classifier could be generated from a small portion of the input dataset provided they outline the whole dataset approximately. The advantages of the approach lies in the fact that the smaller the input dataset is, the fewer SVs would be yielded and that it would require less CPU time and memory .Hence, k-mean clustering is introduced to choose the actual training set. k-mean clustering is applied respectively to the two groups into which the input datasets are divided according to the values of the output data y_i in order to generate two sets of centers. Centers are chosen such that the points are mutually farthest apart, which would well reflect the relative position of point-clusters of input dataset and thus characterize the outline of the full dataset. To achieve an optimal k, i.e. the number of centers, which would describe the full input dataset well, a portion of data will be removed as tuning set to adjust the number of centers to reach the best training precision.

As a result, the procedures for determining k and the set of smaller size can be summarized as follows:

Step 1. Remove a certain portion of an input set as the tuning set and divide the input dataset into two groups according to their labels y_i.

Step 2. Start with a small k, which is around 5% of the whole input set.

Step 3. Apply k-mean algorithm to the two groups respectively to produce a center set as the real dataset to train SVM.

Step 4. Apply the standard SMO algorithm to the training set in order to produce a classifier.

Step 5. Compute the correctness of the classifier on the tuning set

Step 6. If the correct rate of the tuning set is small enough, terminate the loop. Otherwise, increase k and continue from *Step 3*.

3.2 The Second Method Using k-Mean to Reduce the Number of SVs in the Decision Function of SVM Classifier

The second modification to the standard SMO aims at simplifying the decision function of the SVM classifier to strength its generalization capability. It has been noted that the number of SVs that nonlinear separable datasets generate makes up a large proportion of the input set, which would result in a high risk of poor performance on testing examples thus weak generalization capability according to inequality (1). To avoid problems mentioned above, k -mean clustering is employed after the training phase of SVM to suppress the number of SVs.

In the decision function of the SVM classifier, the normal vector of the optimal hyperplane is described by the kernel expansion.

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) = \sum_{i=1}^{N} \lambda_i \phi(x_i) \cdot \tag{11}$$

Now we wish to find a new solution:

$$w^* = \sum_{i=1}^{m} \beta_i \phi(s_i) , \tag{12}$$

so that the kernel expansion would be shorter, i.e. $1 \le m << N$ and well approximate the original expansion.

To simplify the problem, set both m and β as 1 and the problem of finding the new expansion of the normal vector can be formulated as the following optimization task:

$$\begin{aligned} s &= \arg\min_{s'} \| w - w^* \|^2 \\ &= \arg\min_{s'} \| \phi(s) - \sum_{i=1}^{N} \lambda_i \phi(x_i) \|^2 \\ &= \arg\min_{s'} k(s,s) - 2\sum_{i=1}^{N} \lambda_i k(s, x_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j k(x_i, x_j) \end{aligned} \tag{13}$$

For Gaussian radial basis function (RBF) as shown in the equation below:

$$K(x,y) = \exp\left(-\| x-y \|^2 / \left(2\sigma^2\right)\right) , \tag{14}$$

optimization task (13) leads to:

$$s = \arg\max_{s'} \sum_{i=1}^{N} \lambda_i \exp(-\| x_i - s \|^2 / \sigma^2) . \tag{15}$$

For $m \ge 1$, the problem described by Equation (12) with RBF kernel can be converted into m optimization tasks of (15) which aims at finding an input vector s of the input dataset .

To solve the optimization task (15), k -mean is again used to cluster the points of the input dataset. And the algorithm of finding a shorter kernel expansion can be summarized as follows:

Step 1. Start with a small k , around 5% of the size of the input dataset.

Step 2. Apply the k -mean algorithm to yield k centers.

Step 3. Employ the set of these k centers as the actual input dataset and pick up the center which solve the optimization task (15)

Step 4. Compute the deviation between the two expansions of normal vector of the hyperplane

Step 5. If the standard deviation is small enough, terminate the loop. Otherwise increase k and start from *Step 2*.

4 Experimental Results

To verify the effectiveness and efficiency of the novel SMO combined with k -mean clustering, we use Riply's training dataset [14], which contains 250 points, and checkerboard's dataset [15] of 1000 points to test the proposed algorithms. All experiments are conducted on a platform of a machine with a Pentium 4 2.6GHz processor and 265 megabytes of memory.

4.1 Experiments of Combing the First Method with Standard SMO Algorithm

The first experiment on Riply's dataset uses Gaussian radial basis function (RBF) as the kernel function. Figure 1 depicts the decision boundaries of standard SMO with the red solid line and the novel SMO using k -mean with the blue line. The comparison of the novel SMO with the original one is demonstrated by Table 1. Parameter setting for standard SMO is C =30 in Equation (9) and σ = 1 in Equation (14) after model selection. For the SMO using k -mean, C =5 and σ =1 in Figure 1.

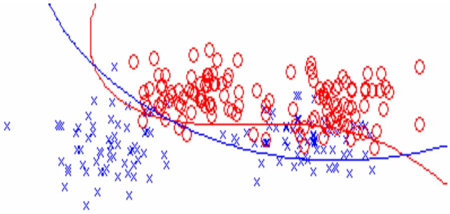

Fig. 1. Decision boundaries built from two SMO algorithms with 8 points (k =4)

Figure 1 illustrates two decision boundaries which bear several similarities to each other. However, the SMO using k -mean clustering shown in Figure 1 only employs 5 SVs while the standard SMO 94 SVs according to Table 1. It shows that the training of SVM has been sped up with the combination of k -mean clustering.

The second experiment, using Gaussian RBF kernel, is to classify the checkerboard dataset. Figure 2 illustrate the training results of the SMO using k -mean with the training set of only 16 points.

Table 1. Performance comparison between the two SMO algorithms 1=standard SMO ; 2 =SMO using k -mean

	k	Training Error	Testing Error	Number of SVs	**Time (CPU sec)**
1		0.128	0.096	91	**1.178**
2	4	0.148	0.094	5	**0.016**
	8	0.192	0.166	10	**0.016**
	16	0.140	0.099	22	**0.031**
	32	0.148	0.099	31	**0.031**
	64	0.128	0.095	57	**0.063**

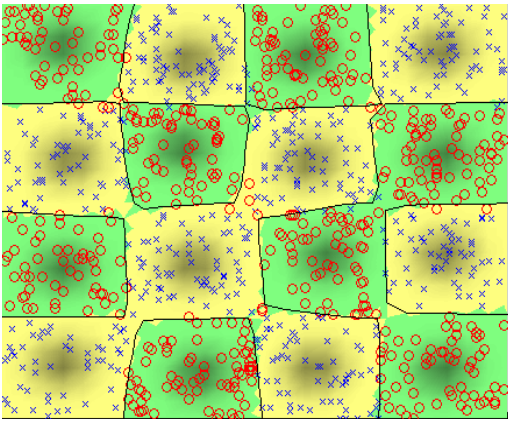

Fig. 2. Performance of SMO using k -mean on checkerboard with 16 training points (k =8) with the parameter setting: C =20 and σ =8

Table 2. Performance comparison between the two SMO algorithms 1=standard SMO; 2 =SMO using k -mean

	k	Training Error	Number of SVs	**Time (CPU sec)**
1		0.000	285	**226.2**
2	8	0.034	16	**0. 11**
	16	0.105	32	**0. 15**
	32	0.112	61	**0. 67**
	64	0.044	118	**0.75**
	128	0.030	192	**0.92**

From Table 2, it can be drawn that the classifier in Figures 2 which gives pretty good a representation of the checkerboard dataset are built on only 1.6% input data and 96.6% input data are classified correctly. A deeper comprehension of the advantages of the SMO using k -mean over the standard SMO can be also seen from the computational time.

4.2 Experiments of Combing the Second Method with Standard SMO Algorithm

The first experiment to verify the effectiveness of the second approach is implemented on Riply's dataset uses Gaussian radial basis function (RBF) as the kernel function with parameters: C =20 and σ =1.

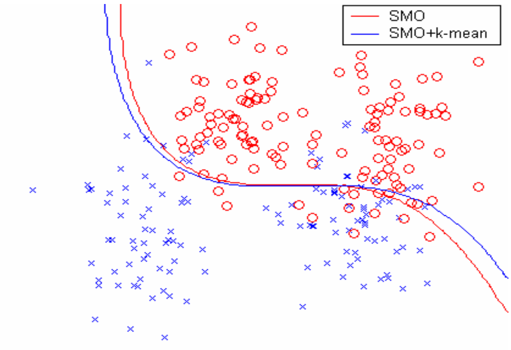

Fig. 3. Decision boundaries built from two SMO algorithms (k =8)

With Figure 3, it is shown that the decision boundaries built from two SMO algorithms are very similar to each other. However, the SVM classifier built with the second method to suppress the number of SVs only employs 8 SVs while the classifier built with standard SMO has 88 SVs.

The second experiment is to classify the checkerboard dataset using Gaussian RBF kernel. After the second method is applied, the normal vector of the hyperlane is spanned with 91 SVs while the original expansion of the normal vector has 273 SVs. The employment of the second method to reduce the number of SVs decreases the expectation value of the probability of committing an error on a test example and enhances SVM's generalization capability. Figure 4 illustrates the training results of the SVM using the second method with k =91.

5 Conclusions

This paper proposes and implements two methods which are intensely involved with k -mean clustering algorithm to suppress the number of SVs. It is shown with experiments that the first method of integrating k -mean clustering into the standard SMO algorithm significantly speeds up the training process and greatly decrease the

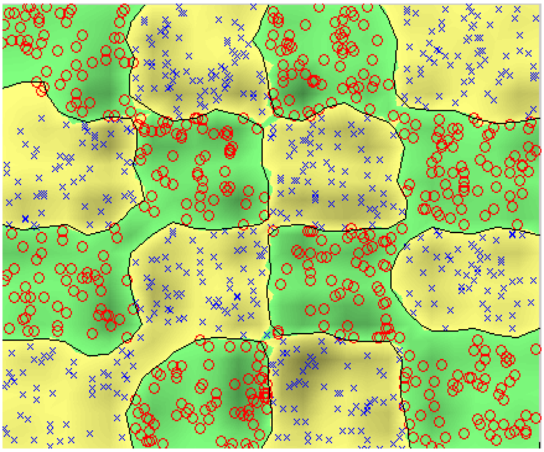

Fig. 4. Performance of the SMO using the second method (k =91)

number of SVs and the second method of combining k -mean clustering with standard SMO makes the number of SVs which span the decision function of the SVM classifier smaller and improves SVM's generalization capability. Future works involves applying the two methods to more real-world problems and modifying k -mean clustering algorithm so that the optimum value for the number of centers can be found.

Acknowledgements

The research is supported by the National Science Foundation of China (Nos.60472111 and 60405002), RGC Project No.CUHK 4170/04E, RGC Project No. CUHK4205/04E and UGC Project No.AoE/E-01/99.

References

1. V.,Vapnik: The nature of statistical learning theory. Springer Verlag, (1995)
2. D.S. Huang, Horace, H.S.Ip, Law Ken, C.K., Zheru Chi: Zeroing polynomials using modified constrained neural network approach. IEEE Trans. On Neural Networks, vol.16, no.3 (2005) 721-732
3. D.S. Huang, Horace, H.S.Ip, Zheru, Chi: A neural root finder of polynomials based on root moments. Neural Computation, Vol.16, No.8 (2004) 1721-1762
4. D.S. Huang: A constructive approach for finding arbitrary roots of polynomials by neural networks. IEEE Transactions on Neural Networks, Vol.15, No.2 (2004) 477-491
5. D.S. Huang: Systematic Theory of Neural Networks for Pattern Recognition. Publishing House of Electronic Industry of China, Beijing (1996)
6. N. Cristianini, J. Shawe-Taylor: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press(2000)

7. Bing-Yu Sun, D.S. Huang, Hai-Tao Fang: Lidar signal de-noising using least squares support vector machine. IEEE Signal Processing Letters, vol.12, no.2 (2005) 101-104
8. Bing-Yu Sun, D.S. Huang: Least squares support vector machine ensemble. The 2004 International Joint Conference on Neural Networks (IJCNN2004), Budapest Hungary (2004) 2013-2016.
9. T.,Joachims.: Making large-scale support vector machine learning practical. Advances in kernel methods: support vector learning. MIT Press (1999) 169-184
10. J. Platt.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances in kernel methods: support vector learning, MIT Press, (1999) 185-208
11. Anil K.Jain, Richard C. Dubes: Algorithms for Clustering Data. Prentice Hall (1988)
12. B. Scholkopf, A.J.,Smola: Learning with Kernels.The MIT Press, MA (2001)
13. J. H. Friedman, F. Baskett, and L. J. Shustek: An algorithm for finding nearest neighbours. IEEE transactions on Computers C-24 (1975)1000-1006
14. B. D. Riply: Neural networks and related methods for classifications. J.Royal Statistical Soc. Series B,56 (1994)409-456
15. ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/checker