Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

AN EFFECTIVE ALGORITHM FOR INVERSE PROBLEM OF SVM BASED ON MM ALGORITHM

JIE ZHU[1] ,RUN-YA LI[2] ,SHU-FANG WU[3] ,SONG JI[2] MAN LI[1]

[1]Department of Information Management,The Central Institute for Correctional Police, Baoding 071000, China
[2]Department of information Engineering,China University of Geosciences Great Wall College, Baoding 071000, China
[3] Information Engineering Department,HeBei Software Institue, Baoding 071000, China
E-Mail:arthurzhujie@yahoo.com.cn

Abstract:

This paper investigates an effective algorithm for inverse problem of support vector machines. The inverse problem is how to split a given dataset into two clusters such that the margin between the two clusters attains maximum. However the training time for inverse problem of SVM is incredible. Clustering is a feasible way to simplify the process of it, but it is difficult to estimate the number of the clusters. In this paper, we design a margin-merging cluster algorithm to solve this problem. We compare our approach with the k-means solution in terms of accuracy loss and training time. Simulations show that the proposed algorithm can solve it efficiently.

Keywords:
Support vector machines; Margin-merging(mm) Cluster; Precision

1. Introduction

Support vector machines (SVMs) are a classification technique of machine leaning based on statistical leaning theory [1, 2]. Considering a classification problem with two classes, SVMs are to construct an optimal hyper-plane that maximizes the margin between two classes. According to Vapnik statistical learning theory [1], the maximum of margin implies the extraordinary generalization capability and good performances of SVM classifiers. The Theory of inverse problem of SVM has been referred in [3], it has been used in the decision tree generation successfully. This paper aims to use a more efficient algorithm called margin-merging cluster to decrease the time complexity and offset the disadvantage brought by k-means algorithm.

The training time is a serious obstacle in the enumerate way. Clustering have been submitted to enhance the training performance, a problem followed by the decreasing of the time complexity is how to determine the number of the clusters., also we can't make sure the margin gained is the maximum one.

The investigation of the margin-merging cluster is motivated by making the process feasible and efficient. Due to the relationship between the margin of SVM and the minimal distance between to clusters, the margin-merging cluster algorithm may be consider one of the best way.

This paper has the following organization. Section 2 briefly reviews the concept of support vector machines. Section 3 proposes the inverse problem of SVM and designs a margin-merging clustering algorithm to solve this problem. Section 4 gives some simulation to demonstrate the feasibility and effectiveness of the new algorithm. And the last section briefly concludes this paper.

2. Support vector machines

Support Vector Machines (SVM) are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory. This learning strategy, introduced by Vapnik and co-workers, is a very powerful method that in the few years since its introduction has already outperformed most other systems in a wide variety of applications. SVM is based on the idea of hyper-plane classifier, or linear separability.[4]

A set of training sample $(x_1, y_1), (x_2, y_2).....(x_n, y_n), x \in R^m, y \in \{+1, -1\}$ separated by the hyper-plane

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

$$(w \cdot x) + b_0 = 0 \tag{1}$$

If the vector set can be separated by the hyper-plane without error, and the distance between the hyper-plane and their nearest vector is maximized, we can say the vector set can be separated by the optimal separating hyper-plane. In the linear SVMs, it can be described as Figure 1.

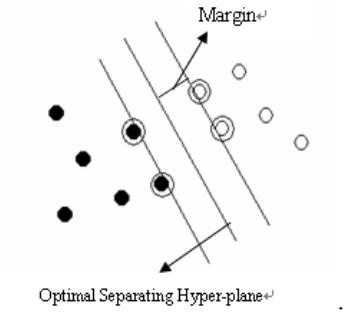

Figure 1. SVM

The separating hyper-planes can be described as follow:

$$y_i[(w \cdot x_i) + b_0] \geq 1 \quad i = 1, 2, 3 \ldots n. \tag{2}$$

The optimal separating hyper-plane can be described as:

$$(w \cdot x_i) + b_0 = 0 \tag{3}$$

The margin of the separation is equal to $1/\|w\|$. Solving the problem to construct the optimal separating hyper-plane as follow:

$$\min \phi(w) = \frac{1}{2}(w \cdot w)$$
$$\textit{subject to}$$
$$y_i[(w \cdot x_i) + b] \geq 1 \quad i = 1, 2, 3 \ldots n. \tag{4}$$

3. Inverse problem of support vector machines and its solutions

For a given dataset of which no classes labels are assigned to instance, we can randomly split the dataset into two subsets. Suppose that one is the positive instance subset and the subsets can be separated by the way of SVM. Then we can get the margin. Obviously the calculated margin depends on the random split of the dataset.

It is an optimal problem. We mathematically formulate it as follow. Let $S = \{x_1, x_2, \ldots, x_n\}$ be a dataset and $x_i \in R^m$ for $i = 1, 2, \ldots, n$, $\Omega = \{f \mid f \textit{ is a function from } S \textit{ to } \{1, -1\}\}$ .Given a function $f \in \Omega$, the dataset can be split to two subsets and then the margin can be calculated by SVM. We denote the calculated margin (the function) by $M \arg in(f)$. Then the inverse problem is formulated as

$$Maximum_{f \in \Omega}(M \arg in(f)) \tag{5}$$

Due to the exponentially increased complexity, it is not feasible to enumerate all possible function in Ω for calculating their margins [3]. It is difficult to give an exact algorithm for solving the optimization problem (5).We also try to solve this problem in diminishing the time complexity of SVM through hierarchical clustering algorithm[5],but the enumeration problem still exist, diminishing the enumeration times or change it into another problem is the key to this problem.

3.1 Solution based on k-means

For a given dataset, if we try to enumerate all the possible partition, 2^n kinds of partition should be considered if the number of the data is n. It is obviously that the time complexity depends on the number of the dataset. Diminishing the number n looks a good way, so we introduce clustering into this problem. First we separate the data into k clusters. Every data in the same cluster has the same label during splitting, second enumerate all the possible cases. The final number of the possible cases is k.. If k<<n and we can get the maximal margin the process will greatly be simplified. Larger the scale of k is faster the process will be, but the possibility of failing to get the maximal margin increases. We need a reasonable balance between k and the accuracy.

3.2 Solution based on margin-merging cluster

The algorithm based on k-means makes the possible partition from 2^n to 2^k, if we can decrease the number k, this process will be simplified again. Merging cluster may be helpful to our research, clusters are merged through some relations between clusters, margin is an important result in SVM, Figure2 shows the relationship between the minimal distance (L) between two data in different clusters and margin that $L \geq 2 * m \arg in$, both of them are usually estimated in the feature space by kernel. Data points in the same circle belong to one cluster.

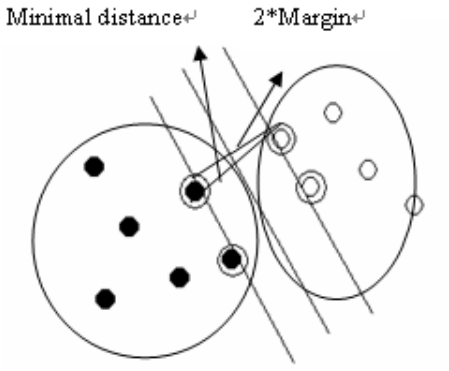

Figure 2. Relationship between margin and minimal distance

Our algorithm can be described as follow:

Step1: One input parameter is selected: margin test times T, number of clusters N

Step2: The K-means clustering algorithm is used on the original data, the data are separated into N clusters.

Step3: Calculating the minimal distance between each two clusters. Here we use the minimal distance between two data in different clusters to represent the minimal distance between two clusters. i=0

Step4: i=i+1. SVM classifier is built and we can get the margin m_i. During this process, data in the same clusters are given the same label randomly.

Step5: If all the minimal distance are larger than $2m_i$ go to (4).If i<T comparing all the minimal distance with $2m_i$, merging every two clusters into new one if the minimal distance is smaller than $2m_i$, else go to(7)

Step6: Calculating the minimal distance between new clusters and other clusters.

Step7: Enumerating all the possible cases and then get the maximal margin.

Our algorithm overcomes two disadvantages of the algorithm based on k-means, first we get a minimal number of the clusters and its clustering result, in this way we can decrease the running time on enumerating all possible cases after clustering. Second we overcome the possible that missing the maximal margin. K-means may make the real hyper-plane through one cluster figure 3. In SVM, the data were mapped into a higher dimensional feature space and the distribution of data was changed, so the clustering vectors and distances were calculated in the feature space during the process of calculating the distance between the clusters[6].In order to fast the process of solving SVM, clustering algorithm can also be used[7].

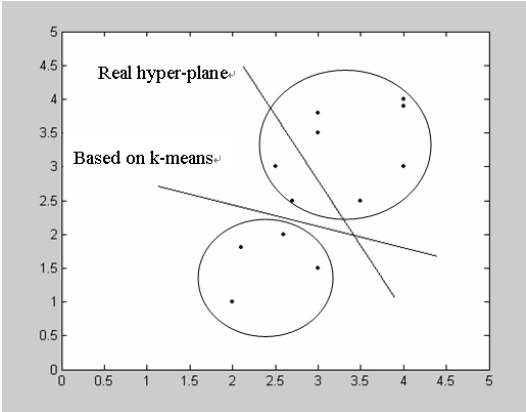

Figure 3. Result after k-means

It is obviously that k-means fails to get the hyper-plane, because the k-means is only a type of clustering algorithm, it doesn't combine the clustering result with the margin.

4. Numerical example

To verify the performance of the margin-merging cluster algorithm, we construct a dataset with 200 2-dimensional points; Figure 3 shows the distribution of the 200 points. Respectively we use.

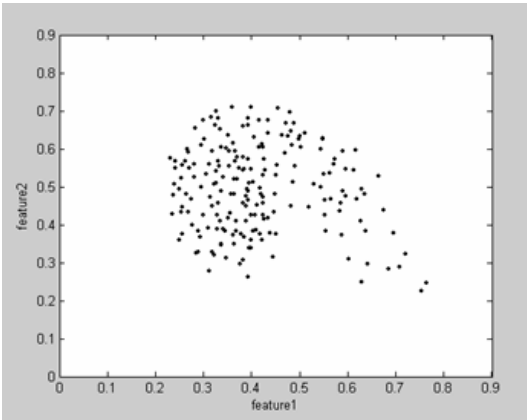

Figure 4. Sample distribution

The proposed margin-merging cluster algorithm and k-means algorithm to acquire the maximum margin, the result is shown in table 1.

Table 1. Experiment on the feature space

Algorithm	Cluster number	Running time	margin
K-means Algorithm	6	297.812	3.464
Margin-merging cluster Algorithm	6	426.751	5.686

From Table 1 we can see that running time of the proposed new algorithm is nearly 1.3 times bigger than the k-means algorithm, but the performance of the new algorithm is better than the k-means one, because it is impossible to miss the largest margin.

The clustering result is shown as Figure 5 different clustering results make the margin different. This is the reason why we try to improve some clustering algorithm to solve this problem. Table 1 shows the better performance of margin-merging cluster algorithm.

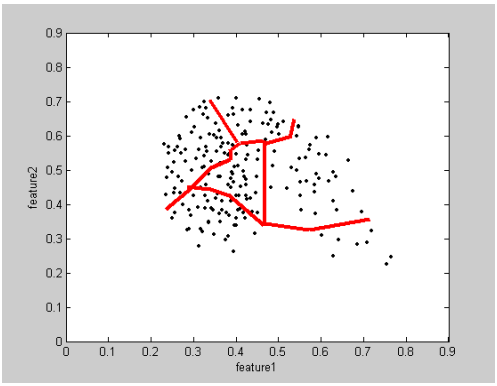

Figure 5. A Margin-merging cluster result

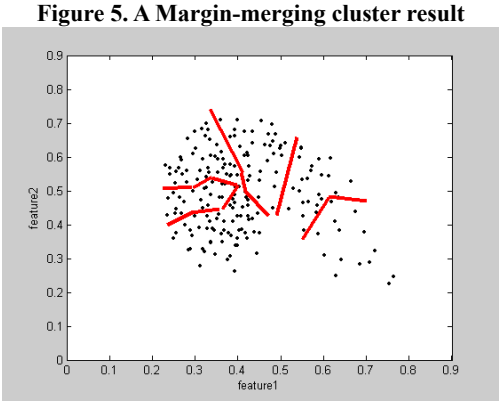

Figure 6. K-means result

The following table 2 describes the performance of the new algorithm.

Table 2. Performance of the new algorithm.

Experiment no	Number of clusters	Margin	Total time	Testing time
1	120	1.5379	13.315	1
2	70	3.8643	22.892	1
3	25	4.8561	50.859	3
4	20	5.0565	59.512	1
5	8	5.6864	70.555	2
6	6		126.513	4
7	6		426.751	

The two parameters given are T=4 and N=120. In the first experiment the total time 13.31546s means the sum of time include clustering the 200 data into 120 clusters, obtaining a random margin and calculating the minimal distance between the clusters. We have to gain a margin to be a standard of margin-merging cluster algorithm, but not all the margin can help us to merge the clusters. We have to search a proper one again and again, the time complexity of the SVM is very high, so a lot of time is wasted in obtaining margin. The third experiment our testing time is 3, the sixth experiment after 4 times testing we have to stop, and enumerate all the possible cases. Step 7 means the process we enumerate all the possible cases.

In this new algorithm every time we have to calculate the minimal distance between clusters that have been changed and other clusters. Will this process cost so much time? Figure 5 gives us an answer.

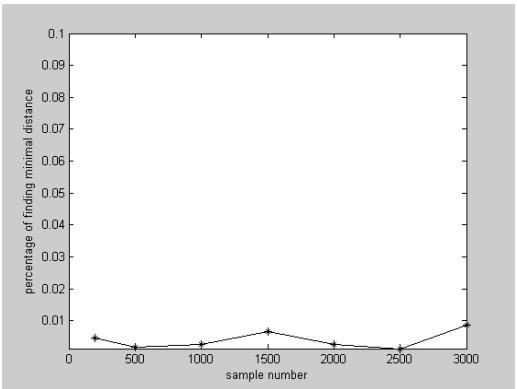

Figure 7. Percentage of finding minimal distance change with the increase of sample

In order to improve our supposal we did some experiments, we choose 7 groups of data, the number of these data scale is 200, 500, 1000, 1500 ,2000, 2500 ,3000. we use the time of finding minimal distance between(TF) and the total running time(TT) to describe this problem, TF/TT means the percentage of finding minimal distance to the total time. Figure 5 shows that all the percentage is lower than 1%. This experiment improves that the number

of the data has little influence to the running time.

5. Conclusions

Motivated by designing a more efficient clustering algorithm to solve inverse problem of SVM, this paper proposes a margin-merging clustering algorithm based on the relationship between the margin and minimal distance between clusters. This algorithm is practical and efficient.

References

[1] V. N. Vapnik, Statistical learning theory, New York, Wiley, 1998, ISBN:0-471-03003-1.

[2] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 2000, ISBN: 0-387-98780-0.

[3] Xi-zhao Wang, Qiang He, De-Gang Chen, Daniel Yeung, A genetic algorithm for solving the inverse problem of support vector machines, Neurocomputing 2005: 225-238.

[4] Awad, M,Khan, L., Bastani, F., I-Ling Yen, An effective support vector machines (SVMs) performance using hierarchical clustering, Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence 2004: 663-667.

[5] Latifur Khan,Mamoun Awad,Bhavani Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, The VLDB Journal, 2007(16):507-521.

[6] Zhang Qilong, Shan Ganlin,Duan Xiusheng, Weighted Support Vector Machine Based Clustering Vector,Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 2008(1):819-822.

[7] Jair Cervantes,Xiaoou Li,Support vector machine classification for large data sets via minimum enclosing ball clustering, Neurocomputing,2008(71):611-619