UNSUPERVISED CLUSTERING BASED REDUCED SUPPORT VECTOR MACHINES

Zheng Songfeng, Lu Xiaofeng, Zheng Nanning, Xu Weipu

The Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, P.R.China 710049

ABSTRACT

To overcome the vast computation of the standard support vector machines (SVMs), Lee and Mangasarian proposed reduced support vector machines (RSVM). But they select 'support vectors' randomly from the training set, and this will affect the test result. In this paper, we select some representative vectors as support vectors via a simple unsupervised clustering algorithm, and then apply the RSVM method on these vectors. The proposed method can get higher recognition accuracy with fewer support vectors compared to the original RSVM, with the advantage of reducing the running time significantly.

1. INTRODUCTION

Support vector machines[1] (SVM) classifier can automatically search the support vectors with stronger classification ability to maximize the margin between the classes, thus it has excellent generalization and higher classification accuracy.

The standard SVM algorithm[1] gets the 'support vectors' via solving a quadratic programming problem, whose time complexity is exponentially. Thus, for large-scale training sets, the computation of the standard SVM is unconceivable. Also, for large-scale training sets, the number of support vectors may be very large, and this leads to time-consuming in the judging process, thus it is impractical in the real-time systems. So, for large-scale training sets, it is necessary to amend the standard SVM algorithm.

To reduce the computation of the standard SVM, Lee and Mangasarian proposed the reduced SVM (RSVM)[2] in 2001, whose fundamental thought is to choose about 1%-10% samples from the training set, regarding them as 'support vectors', then using the whole training set as constrain to get a less-scale quadratic programming problem. The computation for solving this new quadratic programming problem is far less than the standard SVM. But in this method, the 'support vectors' are chosen randomly, lack of representative, and this will lead to the unstable results.

In general, we expect that the more representative the 'support vectors' are, the better the RSVM performs. Unsupervised clustering method can give out a series of clustering centers, which doubtless can represent the training set in a sense. We can regard the clustering centers as 'support vectors', and then apply RSVM algorithm on these samples. We tested our thought via experiments, and got better results compared with the original RSVM algorithm

2. REDUCED SUPPORT VECTOR MACHINES[2,3]

Suppose that the training set is $(x_i, y_i), i=1,2,\ldots,l$, where $x_i \in R^n$, $y_i \in \{-1,1\}$, traditional SVM algorithm is summarized as the following optimization problem:

$$\min_{w,\xi} \quad \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i \qquad (1)$$
$$s.t. \quad y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i$$

the time complexity of solving this optimization problem is exponential. For a large-scale training set, the computation for solving this optimization problem is unconceivable; and in this case, the number of support vectors may be very large, and this leads to a large computation for the classification super-plane.

In [3], (1) is rewritten as:

$$\begin{aligned} &\min_{w,\xi,b} \quad \frac{1}{2}(w^T w + b^2) + C\sum_{i=1}^{l} \xi_i^2 \\ &s.t. \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \end{aligned} \quad (2)$$

And [3] points out that after this change, the object function can get stronger convexity, and the solutions of (2) and (1) don't differ much.

The weight of the optimal super-plane is:

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \quad (3)$$

where α_i are the *Lagrangian* multipliers according to the constrains in (2).

Thus:

$$\begin{aligned} w^T w &= \sum_{i=1}^{l}\sum_{j=1}^{l} y_i \alpha_i \phi(x_i)^T y_j \alpha_j \phi(x_j) \\ &= \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i Q_{ij} \alpha_j = \alpha^T Q \alpha \end{aligned} \quad (4)$$

$Q_{ij} = y_i K(x_i, x_j) y_j$, here $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel function. We have:

$$\begin{aligned} y_i w^T \phi(x_i) &= \sum_{j=1}^{l} y_i y_j \alpha_j \phi(x_j)^T \phi(x_i) \\ &= \sum_{j=1}^{l} Q_{ij} \alpha_j = (Q\alpha)_i \end{aligned} \quad (5)$$

substituting (4), (5) into (2), we get a new optimization problem:

$$\begin{aligned} &\min_{\alpha,b,\xi} \quad \frac{1}{2}(\alpha^T Q \alpha + b^2) + C\sum_{i=1}^{l} \xi_i^2 \\ &s.t. \quad Q\alpha + by \geq e - \xi \end{aligned} \quad (6)$$

where e is an arbitrary dimensional column vector, whose elements are all 1.

The main thought of RSVM is to randomly choose m vectors from the training set, and suppose that the coefficients of the classification supper-plane can be expressed as:

$$w = \sum_{i \in R} y_i \alpha_i \phi(x_i) \quad (7)$$

where R is the set of the subscription of the chosen vectors. Substitute (7) into (2), by the similar deducting as above, we can get an optimization problem similar to (6), but the number of main variables (α) decreased to m:

$$\begin{aligned} &\min_{\overline{\alpha},b,\xi} \quad \frac{1}{2}(\overline{\alpha}^T Q_{RR} \overline{\alpha} + b^2) + C\sum_{i=1}^{l} \xi_i^2 \\ &s.t. \quad Q_{:,R}\overline{\alpha} + by \geq e - \xi \end{aligned} \quad (8)$$

where $\overline{\alpha}$ is the collection of all possible α_i, $i \in R$, Q_{RR} is a square matrix composed of the rows and columns of $\boldsymbol{Q}$ corresponding to the chosen vectors, and $Q_{:,R}$ is a matrix composed of the columns of $\boldsymbol{Q}$ corresponding to the chosen vectors.

Solving (8) and (7), we can get the coefficients of the classification super-plane. Compared to (6), the computation of solving (8) is far less, because the chosen samples only count 1%-10% of the whole training set. Note that in the constrains of (8), the matrix $Q_{:,R}$ also uses the unchosen samples, thus the information contained by the unchosen samples is not abandoned, but is used adequately.

3.UNSUPERVISED CLUSTERING BASED RSVM

In the RSVM algorithm above, the 'support vectors' are chosen randomly, thus its classification results are affected by the randomness greatly. In general, we expect that the more the 'support vectors', the better the result.

In fact, we require that the 'support vectors' can represent the training set adequately, the less number the better, and they are not necessarily in the training set. So we can choose some representative points as 'support vectors' at the first step, and then use these points to train the RSVM.

We use a simply unsupervised clustering algorithm [4] to find the representative points. The algorithm is expressed as following:

Suppose that the sample set for clustering is X

$= \{x_1, x_2, \ldots, x_m\}$, $x \in R^n$, and the clustering radius is r.

Step 1. $C_1 = \{x_1\}$, $O_1 = x_1$, $Cluster\_num = 1$, $Z = \{x_2, \ldots, x_n\}$

Step 2. *if* $Z = \Phi$ *, then STOP*

Step 3. for a sample $x_i \in Z$, choose the clustering center O_j closest to x_i from the existing clustering centers, i.e:

$$O_j = \arg_j \min_{k=1}^{Cluster\_num} d(x_i, O_k)$$

Step 4. if the Euclidian distance $d(x_i, O_j) < r$, add x_i into class C_j, i.e: $C_j = C_j \bigcup \{x_i\}$, the clustering center of this class is adjusted to the mean value of all the samples in this class, go to Step 6.

Step 5. if $d(x_i, O_j) > r$, add a new class $Cluster\_num = Cluster\_num + 1$, $C_{Cluster\_num} = \{x_i\}$, $O_{Cluster\_num} = x_i$

Step 6. $Z = Z - \{x_i\}$, go to Step 2

If our goal is to cluster the samples to K classes, then the time complexity of this algorithm is $O(Km)$.

We can give out our main algorithm as follows:

Step 1. for the positive and negative sample sets of the training set, apply the unsupervised clustering algorithm respectively, and get two sets of clustering centers $\{O_1^+, O_2^+, \ldots\ldots, O_p^+\}$ and $\{O_1^-, O_2^-, \ldots\ldots, O_n^-\}$, their labels are denoted as 1 and –1 respectively.

Step 2. treat these clustering centers as 'support vectors', substitute them to the optimization problem as (8), solve (8) can get the coefficients α corresponding to the 'support vectors'.

Step 3. for a sample x, use the following equation to judge its class

$$f(x) = sign\left(\sum_{i=1}^{p} \alpha_i^+ K(x, O_i^+) - \sum_{i=1}^{m} \alpha_i^- K(x, O_i^-) + b\right) \quad (9)$$

4.EXPERIMENTAL RESULTS

To verify the validity of our method, we classified two data sets by our method and the method in [2] respectively, and compared the test results. The platform of our experiments is MATLAB6.1, the memory of the computer is 256M and CPU is Pentium3 1GHz.

The first dataset is Checkerboard dataset, which contains 1000 data sampled randomly from 16 black and white squares of a Checkerboard. The kernel function we used is Gaussian function $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{400})$, and the parameter C in (8) is 100. The test results as listed in table1 and table2. We could conclude from the tables that in order to get the similar classification accuracy, the number of 'support vectors', the time spent on solving optimization problem and the total running time our algorithm needs are all far less than those of the original RSVM algorithm.

The second dataset is the face database provided by the AI lab of MIT. The training set consists of 6,977 samples, containing 2,429 face images and 4,548 nonface images; the test set consists of 24,045 samples, containing 472 face images and 23,573 nonface images. The kernel function we used is $K(x_i, x_j) = (1 + x_i \cdot x_j)^2$. We set the clustering radius for the positive sample and negative samples to be 4.3 and 5.7 respectively, and we got 98 positive clustering centers and 193 negative centers, i.e. the number of 'support vectors' is 291. When we use the original RSVM algorithm, we randomly took 291 samples from the training set as 'support vectors'.

To compare the results, we adjust the threshold value b in (9) from $-\infty$ to $+\infty$, and we can get the classification accuracy rates and the false positive rates of the two algorithms respectively. We figured out the test results in figure 1, and got Receiver Operator Characteristic (ROC) curve. From the figure, we could conclude that under the same false positive rate, our algorithm can get higher classification accuracy rate.

Table 1. Experimental results gotten by our algorithm

Clustering radius (r+ = r-)	70	60	50	30	15
The number of "support vectors"	12	15	18	43	120
The time spent on solving (8)	1.98	2.44	3.11	8.31	18.45
The total running time(s)	7.70	9.07	10.33	23.73	64.24
Classification accuracy (%)	78.5	89.8	95.3	96.4	96.9

Table 2. Experimental results gotten by original RSVM algorithm

The number of "support vectors"	30	40	70	150	210
The time spent on solving (8)	4.31	5.77	9.67	20.73	30.17
The total running time(s)	16.27	21.28	36.48	80.09	110.6
Classification accuracy (%)	78.5	89.7	95.2	96.4	96.9

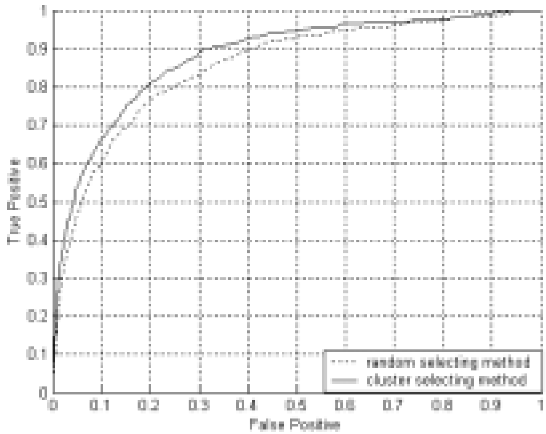

Figure 1. ROC curves of CBCL face database gotten by our algorithm and the original RSVM algorithm

5.CONCLUSIONS

This paper modifies the RSVM algorithm proposed by Lee and Mangasarian. We first use the unsupervised clustering algorithm to choose the representative points of the original training set, then use these points as 'support vectors' to train the RSVM classifier. The experimental results indicate that compared to the original RSVM, our method have advantages both in classification accuracy and running time. But the unsupervised clustering algorithm used here has to be given the clustering radius and it is a problem under investigation.

6. REFERENCES

[1] V.Vapnik. The Nature of Statistic Learning Theory[M]. New York: Springer. 1995

[2] Y.-J. Lee and O.L. Mangasarian. RSVM: Reduced Support Vector Machines. In *proceedings of the First SIAM International Conference on Data Mining*, 2001

[3] K.-M. Lin and C.-J. Lin A study on reduced support vector machines. February 2002. At http://www.csie.ntu.edu.tw/~cjlin/papers.html

[4] LI Xiao-Li, LIU Ji-Min and SHI Zhong-Zhi. A Chinese Web Page Classifier Based on Support Vector Machine and Unsupervised Clustering. CHINESE J. COMPUTERS, Jan. 2001, 62-68