Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

A NEW NU-SUPPORT VECTOR MACHINE FOR TRAINING SETS WITH DUPLICATE SAMPLES

YIN-SHAN JIA[1], CHUAN-YING JIA[2], HONG-WEI QI[3]

[1]School of Information Technology Liaoning University of Petroleum and Chemicals,Fushun 113001,China
[2]Lab of Information and Control,Dalian Maritime University, Dalian 116026, China
[3] Fushun Ethylene Complex, Fushun 113004,China
E-MAIL:jiayinshan@sina.com, chuanyingjia@sina.com,hongweiqi@sina.com

Abstract:

Analyzed theoretically, ν-SVM was found to be over-dependent on each training sample, even if the samples have same value. This dependence would result in more time for training, more support vectors and more decision time. In order to overcome this problem, we propose a new ν-SVM. This new ν-SVM multiplies each slack variable in the objective function by a weight factor, and automatically computes each weight factor by the number of corresponding samples with same value before training. Theoretical analysis and the results of experiments show that the new ν-SVM has the same classification precision rate as the standard ν-SVM and the new ν-SVM is faster than the ν-SVM in training and decision if the training sets have same value samples.

Keywords:

Support vector machines; duplicate samples; weighted support vector machines; machine learning

1. Introduction

Support vector machines（SVMs）are canonized by many researchers and have been applied successfully to many classification problems such as character recognition, speech recognition, face recognition, iris recognition, web classification. The advantages of SVM over conventional classification methods are its higher generalization ability especially when the number of training data is small, its adaptability to various classification problems by changing kernel functions, and its global optimal solution.

The first support vector machine, proposed by Cortes and Vapnik[1][2], is a relative new machine learning methodology based on statistical learning theory. This support vector machine is considered as the standard support vector machine and C-SVM[3]. Its basic idea is to find the hyperplane which can separate data belonging to two classes with maximum margin. This hyperplane is called optimal hyperplane. Figure 1 shows the principle of SVMs.

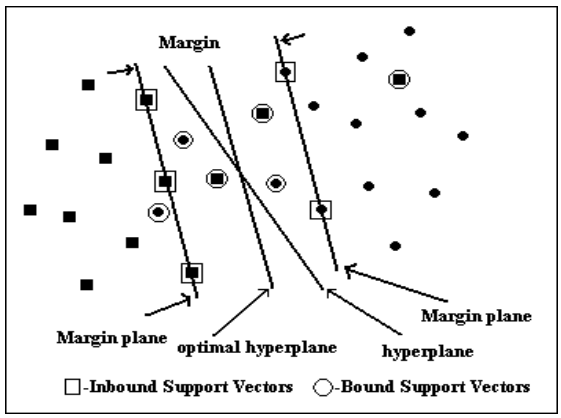

Figure 1. Principles of Support Vector Machines

Many works have been done on SVMs since the first support vector machine was proposed. Schölkopf, et al.[4] proposed ν-SVM to overcome the unclear meaning of parameter C in C-SVM. ν-SVM has been paid more attention because it provides a mean to control the training errors. Chew,et al.[5] proposed a weighed ν-SVM to solve the problem of different error rates of classification resulted from uneven training class sizes. Huang, et al.[6] proposed a weighted C-SVM to reduce the effect of outliers or noises on the training result.

In this paper, a new ν-SVM is proposed. The purpose is to solve the problem that the training result of ν-SVM probably has duplicate support vectors when there are duplicate samples in the training sets. The new ν-SVM is also a weighted ν-SVM. It multiplies each slack variable in the objective function by a weight factor, and automatically computes each weight factor by the number of corresponding samples with same value before training. As a result, the training result of the new ν-SVM has no duplicate support vectors.

2. ν-Support vector machine

2.1. Principle of ν-SVM

Given $Z=\{(x_i,y_i): x_i \in R^N, y_i \in \{+1,-1\}, i=1,...,m\}$ a set of training samples, where each x_i is a data vector, y_i is the label of the class that x_i belongs to. In order to seek the optimal hyperplane that best separates the two classes from each other with the widest margin, we need to solve the following optimization problem:

$$\min \tau(w,\xi,\rho)=\frac{1}{2}w^Tw-\nu\rho+\frac{1}{m}\sum_{i=1}^{m}\xi_i\,, \tag{1}$$

$$\text{s.t} \quad y_i(w^Tx+b)\geq \rho-\xi_i\,, \tag{2}$$

$$\text{and} \quad \xi_i\geq 0, i=1,...,m\,, \tag{3}$$

$$\text{and} \quad \rho\geq 0\,. \tag{4}$$

By Introducing Lagrange multipliers α_i, β_i and δ, we have:

$$L(w,\xi,b,\rho,\alpha,\beta,\delta)=\frac{1}{2}w^Tw-\nu\rho+\frac{1}{m}\sum_{i=1}^{m}\xi_i-\sum_{i=1}^{m}(\alpha_i(y_i(w^T\phi(x_i)+b)-\rho+\xi_i)+\beta_i\xi_i)-\delta\rho. \tag{5}$$

The corresponding dual Lagrangian is:

$$\max W(\alpha)=-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_jy_iy_jk(x_i,x_j), \tag{6}$$

$$\text{s.t.} \quad 0\leq\alpha_i\leq\frac{1}{m}, i=1,...,m\,, \tag{7}$$

$$\text{and} \quad \sum_{i=1}^{m}\alpha_iy_i=0, \tag{8}$$

$$\text{and} \quad \sum_{i=1}^{m}\alpha_i\geq\nu\,. \tag{9}$$

By solving the above dual Lagrangian, we obtain:

$$w=\sum_{i=1}^{m}\alpha_iy_i\phi(x_i)\,. \tag{10}$$

The resulting decision function can be shown as

$$f(x)=\text{sgn}(\sum_{i=1}^{m}y_i\alpha_ik(x_i,x)+b)\,. \tag{11}$$

2.2. Analysis of ν-SVM

For ν-SVM, the Karush-Kuhn-Tucker conditions of the primal problem ((1)-(4)) can be stated:

$$\alpha_i(w^T\phi(x_i)+b-\rho+\xi_i)=0\,, \tag{12}$$

$$\beta_i\xi_i=(\frac{1}{m}-\alpha_i)\xi_i=0\,, \tag{13}$$

$$\delta\rho=0\,. \tag{14}$$

Therefore, there are four cases as follows:

1) If $\alpha_i=0$, according to (13), we obtain $\xi_i=0$. In this case, x_i is correctly classified.

2) If $0<\alpha_i<1/m$, then $y_i(w^T\phi(x_i)+b)-\rho+\xi_i=0$ and $\xi_i=0$. In this case, x_i lies on the margin plane and x_i is called an in-bound support vector.

3) If $\alpha_i=1/m$, then $y_i(w^T\phi(x_i)+b)-\rho+\xi_i=0$ and $\xi_i\geq 0$. If $0\leq\xi_i<\rho$, x_i is correctly classified. If $\xi_i\geq\rho$, x_i is misclassified. In the case of $\xi_i>0$, x_i is located in the margin and is called a bound support vector(BSV).

4) In most cases, ρ is greater than zero, according to (14), we know $\delta=0$. This results in constraint (9) reducing to an equality condition

$$\sum_{i=1}^{m}\alpha_i=\nu \tag{15}$$

Suppose N_{BSV+} and N_{BSV-} are the number of bound support vectors in the positive class and the negative class respectively, and N_{SV+} and N_{SV-} are the number of all kind of support vectors in the positive class and the negative class respectively, and m_+ and m_- are the number of data points in the positive class and the negative class respectively, we have:

$$\frac{2N_{BSV+}}{m}\leq\nu\leq\frac{2N_{SV+}}{m}\,. \tag{16}$$

$$\frac{2N_{BSV-}}{m}\leq\nu\leq\frac{2N_{SV-}}{m}\,. \tag{17}$$

Multiplying (16), (17) by $m/2m_+$ and $m/2m_-$ respectively, and substituting m_++m_- for m, we obtain:

$$\frac{N_{BSV+}}{m_+}\leq\frac{m_++m_-}{2m_+}\nu\leq\frac{N_{SV+}}{m_+}\,. \tag{18}$$

$$\frac{N_{BSV-}}{m_-}\leq\frac{m_++m_-}{2m_-}\nu\leq\frac{N_{SV-}}{m_-}\,. \tag{19}$$

From (18) and (19), we know:

1) Omitting a data point, even if it has the same value as another point, would increase the upper bound of the fraction of bound support vectors and the lower bound of the fraction of support vectors of the class that it belongs to. In other words, omitting a data point would change the optimal hyperplane.

Besides the above characteristics, we can know intuitively that ν-SVM has another characteristic as follows:

2) If the training sets have same value samples, there would be same support vectors. In this case, decision is slower.

3. A new ν-support vector machine

3.1. Description of the new ν-SVM

We have known from the last section that ν-SVM would produce same support vectors if the training sets have same value samples, so we propose a new ν-SVM to solve this problem and decrease the time required for training and decision.

The primal problem in the new ν-SVM is

$$\min \tau(w,\xi,\rho) = \frac{1}{2}w^T w - \frac{m-n}{m}\nu\rho + \frac{1}{m-n}\sum_{i=1}^{m-n} d_i\xi_i , \quad (20)$$

$$\text{s.t} \quad y_i(w^T x + b) \geq \rho - \xi_i , \quad (21)$$

$$\text{and} \quad \xi_i \geq 0, i = 1,...,m-n , \quad (22)$$

$$\text{and} \quad \rho \geq 0 . \quad (23)$$

In above formulations, m is the number of training samples, n is the number of training samples each of which has the same value as a sample in the m-n samples. That is to say, m-n is the number of training samples each of which has a unique value. d_i is the duplicate factor of sample x_i, and $d_i \geq 1$. The term $d_i\xi_i$ in (20) is the error loss resulted from misclassifying x_i.

By Introducing Lagrange multipliers α_i, β_i and δ, we have:

$$L(w,\xi,b,\rho,\alpha,\beta,\delta) = \frac{1}{2}\|w\|^2 - \frac{m}{m-n}\nu\rho + \frac{1}{m-n}\sum_{i=1}^{m-n} d_i\xi_i - \sum_{i=1}^{m-n}(\alpha_i(y_i(w^T\phi(x_i)+b) - \rho + \xi_i) + \beta_i\xi_i) - \delta\rho \quad (24)$$

With the same method as that in ν-SVM, we obtain dual Lagrangian:

$$\max W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j), \quad (25)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{1}{m-n}d_i, i = 1,...,m-n , \quad (26)$$

$$\text{and} \quad \sum_{i=1}^{l}\alpha_i y_i = 0 , \quad (27)$$

$$\text{and} \quad \sum_{i=1}^{l}\alpha_i \geq \frac{m}{m-n}\nu . \quad (28)$$

The decision function of the new ν-SVM is

$$f(x) = \text{sgn}(\sum_{i=1}^{m-n} y_i\alpha_i k(x_i,x) + b) . \quad (29)$$

3.2. Computation of n and the weights d_i

At First, n is initialized to zero. After reading a training sample x_i, the new ν-SVM searches if x_i is the same as any other samples which have been read and stored in the samples list. If there is a same sample x_j which has been stored in the samples list, then n is increased by 1, the corresponding d_j is increased by 1 and x_i is ignored, otherwise x_i is stored in the sample list and d_i is set to 1. After all samples are read, all samples in the samples list are not duplicate.

3.3. Computation complex analysis of the new ν-SVM

Suppose the size of training set is m, and n samples in the set are the same as others. In other words, each of these n samples has the same value as a sample among the m-n samples. And suppose SMO algorithm[8] is used to solve the dual Lagrangian of ν-SVM and that of the new ν-SVM. In training phase, the time complexity of ν-SVM and the new ν-SVM is $O(m^3)$ and $O(n^2+(m-n)^3)$ respectively. Therefore, the more same value samples in the training set, the more time saved if the new ν-SVM instead of ν-SVM is used. From (11) and (29), we can know that if ν-SVM has duplicate support vectors, then the new ν-SVM is faster than ν-SVM in decision.

The space complexity of the new ν-SVM is larger than that of ν-SVM because it has to store the duplicate factors.

3.4. Bounds of the new ν-SVM

With the same method as in section 2.2, we obtain:

$$\frac{1}{m-n}\sum_{x_i\in BSV} d_i \leq \frac{m}{m-n}\nu \leq \frac{1}{m-n}\sum_{x_i\in SV} d_i . \quad (30)$$

Multiplying (30) by $\frac{m-n}{m}$, we obtain:

$$\frac{\sum_{x_i\in BSV} d_i}{m} \leq \nu \leq \frac{\sum_{x_i\in SV} d_i}{m} . \quad (31)$$

Because $\sum_{x_i\in BSV} d_i$ equals the number of bound support vectors and $\sum_{x_i\in SV} d_i$ equals the number of support vectors, the meaning of ν is not changed in the new ν-SVM. In other words, the new ν-SVM has the same bound as ν-SVM.

We can also obtain:

$$\frac{\sum_{x_i\in BSV+} d_i}{m+} \leq \frac{m_+ + m_-}{2m_+}\nu \leq \frac{\sum_{x_i\in SV+} d_i}{m+} . \quad (32)$$

$$\frac{\sum_{x_i\in BSV-} d_i}{m-} \leq \frac{m_+ + m_-}{2m_-}\nu \leq \frac{\sum_{x_i\in SV-} d_i}{m-} . \quad (33)$$

We know $\sum_{x_i\in BSV+} d_i$, $\sum_{x_i\in SV+} d_i$, $\sum_{x_i\in BSV-} d_i$ and $\sum_{x_i\in SV-} d_i$ equal the number of bound support vectors in the positive class, the number of support vectors in the positive class, the number of bound support vectors in the negative class,

the number of support vectors in the negative class respectively, so the upper bound of the fraction of bound support vectors and the lower bound of support vectors in each class are not changed.

4. Experiments

In this section, we use some experiment results to compare ν-SVM and the new ν-SVM presented in this paper. Experiments on ν-SVM were done with libsvm[9], and experiments on the new ν-SVM were done with a new program based on libsvm. Training samples and testing samples are all two dimensional.

Results of training experiments are shown in Table 1 and Results of testing experiments are shown in Table 2.

Table 1. Results of Training Experiment s

ID	*A*	*B*	*ν-SVM*			*New ν-SVM*	
			C	*D*	*E*	*C*	*D*
1	*1000*	*614*	*0.5*	*957*	*572*	*0.28*	*385*
2	*2000*	*805*	*2.8*	*200*	*43*	*1.68*	*157*
3	*3000*	*124*	*6.6*	*138*	*24*	*7.6*	*114*
4	*4000*	*383*	*14.0*	*185*	*35*	*9.5*	*150*

In Table 1, Column A, B, C, D and E denote the number of training samples, the number of duplicate samples, time(seconds) for training, the number of support vectors and the number of duplicate support vectors, respectively.

Table 2. Results of Test Experiment s

ID	*A*	*B*	*C*
1	*50*	*40.6*	*30.7*
2	*50*	*20.3*	*17.2*
3	*50*	*16.5*	*14.8*
4	*50*	*18.9*	*17.1*

In Table 2, Column A, B and C denotes number of test samples, time (milliseconds) for decision with ν-SVM and time(milliseconds) for decision with the new ν-SVM, respectively. Experiments in Table2 use the model generated by corresponding experiments in Table 1.

From results of experiments, we can conclude that the new ν-SVM is faster than ν-SVM in both training and decision if the training set has duplicate samples.

5. Conclusions

We have analyzed the characteristics of ν-SVM and presented a new ν-SVM. Results of theoretical analysis have shown that ν-SVM was over-dependent on each training sample. The new ν-SVM proposed in this paper uses a weight factor to represent the duplicate times of a training sample and compute the weight factor automatically. The new ν-SVM is faster than ν-SVM in training if the training sets have same-value samples, and faster in decision if the results of training of ν-SVM have same-value support vectors. The new ν-SVM has the same classification precision as ν-SVM.

References

[1] Cortes C and Vapnik V., "Support vector networks", Machine learning, Vol. 20, No. 3, pp.273~297,1995.

[2] Vapnik V., The nature of statistical learning theory, Springer- Verlag, New York, 1995.

[3] Schölkopf B. and Smola A J., Learning with Kernels, MIT Press, 2002

[4] Schölkopf B, Smola A J, Williamson R C, et al., "New support vector algorithms", Neural Computation,Vol. 12, No. 5, pp.1207-1245, 2000.

[5] Chew H G, Bogner R E and Lim C C., "Dual-nu support vector machine with error rate and training size Biasing", Proceedings of the 26th International Conference on Acoustics, Speech and Signal Processing, pp. 1269~1272, 2001.

[6] Huang Han-Pang and Liu Yi-Hung, "Fuzzy support vector machines for pattern recognition and data mining",International Journal of Fuzzy Systems, Vol. 4, No. 3, pp.826~835, 2002.

[7] Fletcher,R., Practical Methods of Optimization, John Wiley and Sons Inc., 1987.

[8] Platt J., "Fast Training of Support Vector Machines using Sequential Minimal Optimization", In Schölkopf B, Burges C and Smola A J, eds. Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.

[9] Chang Chih-Chung and Lin Chih-Jen, "LIBSVM: a library for support vector machine", 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.