H K . N C C P
International Journal of Intelligent Information and Management Science
ISSN: 2307-0692 *Volume 3, Issue 1, February 2014*

A Core Set Weighted Support Vector Machines

Shuxia LU, Limin LI
Key Lab. In Machine Learning and Computational Intelligence
College of Mathematics and Computer Science, Hebei University
Baoding 071002, China

Abstract: Classification for large datasets is a classical problem in machine learning. In this paper, we focus on effevtive classification algorithm for large datasets and imbalanced datasets. First, to deal with imbalanced dataset, we define the weight according to the size of positive and negative dataset. Then, a fast learning algorithm on large datasets called a core set weighted support vector machines (CSWSVM) is proposed. In the proposed approach, the corresponding core set (CS) can be solved by employing the core vector machine (CVM) or generalized CVM (GCVM), and then the weighted support vector machines (WSVM) can be used to implement classification for imbalanced datasets. Experimental results on UCI and USPS datasets demonstrate that the proposed method is effective.

Keywords: core vector machine; weight; support vector machine; core set

1. Introduction

Classification for large datasets is a hot issue in current research. It is difficult because many of kernel methods are formulated as quadratic programming (QP) problems [1-4]. The training time complexity of QP is $O(N^3)$ and its space complexity is at least quadratic. To reduce the time and space complexities, a variety of approaches have been proposed in large datasets problem. Typical techniques include the SMO algorithm [5, 6]; the sampling method for kernel methods [7]; matrix decompositions [8]; the core vector machine (CVM) [9, 10]. Tsang et al. proposed the core vector machine (CVM) by utilizing an approximation algorithm for the minimum enclosing ball (MEB) problem in computational geometry, the CVM algorithm achieves an asymptotic time complexity that is linear in N and a space complexity that is independent of N, where N is the size of the training patterns; Hu et al. proposed the maximum vector-angular margin core vector machine[11], by connecting the CVM method with MAMC such that the corresponding fast training on large datasets can be effectively achieved; Chung et al. established the relationship between fuzzy inference systems and CC-MEB by GCVM [12]; Hu et al. proposed a fast learning algorithm for scaling up Minimum Enclosing Ball with total soft margin [13].

In the real world, these training samples are not always balanced. Many researchers have worked to solve this problem so that the classification performance of the majority class and that of minority class are good at the same time. To solve this problem, two methods have been proposed: one is based on sampling method and the other one is based on sample weighting method [14--16]. Sampling method includes: under sampling method and oversampling method. The sample weighting approach to the imbalanced data classification problem is to apply the weights to the training data points.

In this paper, we focus on the large and imbalanced datasets effective classification problem, a weight core vector machine - A core set weighted support vector machines (CSWSVM) approach is proposed. It consists of two stages. The first stage is to obtain the core set of the large training dataset by using the CVM algorithm. In the second stage, we define the weight on the the obtained core set, the WSVM algorithm is utilized to train and yields a decision function for classifying testing patterns. Experiments on large classification datasets also demonstrated that the proposed method has comparable performance with CVM implementations.

The rest of this paper is organized as follows. Section 2 presents the CSWSVM approach. In Section 3, the experimental results on several datasets are reported. Some conclusions are finally given in Section 4.

2. A Core Set Weighted Support Vector Machines (CSWSVM)

2.1. The Generalized Core Vector Machine (GCVM）

In this section, we first review the generalized core vector machine (The generalized CVM, GCVM) algorithm as proposed in [10]. The GCVM utilizes an approximation algorithm for the center constrain minimum enclosing ball (CC-MEB) problem, which will be briefly in-

troduced in Section 2.1.1.

2.1.1. Center Constrain Minimum Enclosing Ball (CC-MEB)

Suppose the training set is denoted by $S=\{x_i \mid x_i \in \ ^n,\ i=1,\mathbf{L},N\}$, the minimum enclosing ball of S (denoted $MEB(S)$) is the smallest ball that contains all the points in S. In this paper, we denote the ball with center $\mathbf{c}$ and radius R by $B(\mathbf{c},R)$. Also, the center and radius of a ball $B(\mathbf{c},R)$ are denoted by $\mathbf{c}_B$ and r_B, respectively. Given an $e>0$, a ball $B(\mathbf{c},(1+e)R)$ is an $(1+e)$-approximation of $MEB(S)$ if $R\le r_{MEB(S)}$ and $S\subset B(\mathbf{c},(1+e)R)$. $j:x_i\to j(x_i)$ denotes the feature map associated with a given kernel k, and $B(\mathbf{c},R)$ is the desired MEB in the kernel-induced feature space Γ.

The MEB problem finds the smallest ball containing all $j(x_i)\in S$ in the feature space. In this section, we first augment an extra $d_i\in R$ to each $j(x_i)$, forming $\begin{bmatrix} j(x_i) \\ d_i \end{bmatrix}$. Then, we find the MEB for these augmented points, while at the same time constraining the last coordinate of the ball's center to be zero (i.e., of the form $\begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}$). The primal form of the center constrain minimum enclosing ball (CC-MEB) problem can be formulated as

$$\begin{aligned} &\min \quad R^2 \\ &s.t. \quad \left\| j(x_i)-\mathbf{c} \right\|^2 + d_i^2 \le R^2, \quad i=1,\mathbf{L},N. \end{aligned} \tag{1}$$

The corresponding dual of (1) is the following QP problem

$$\begin{aligned} &\max \quad \boldsymbol{\alpha}^T(diag(\mathbf{K})+\boldsymbol{\Delta})-\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} \\ &s.t. \quad \boldsymbol{\alpha}^T\mathbf{1}=1, \quad \boldsymbol{\alpha}\ge\mathbf{0}. \end{aligned} \tag{2}$$

where $K=[k(x_i,x_j)]=[j(x_i)^T j(x_j)]$ is the corresponding kernel matrix, and

$$\boldsymbol{\Delta}=[d_1^2,\mathbf{L},d_N^2]^T\ge\mathbf{0}. \tag{3}$$

From the optimal $\boldsymbol{\alpha}$ solution of (2), we can recover R and $\mathbf{c}$ as

$$R=\sqrt{\boldsymbol{\alpha}^T(diag(\mathbf{K})+\boldsymbol{\Delta})-\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}} \tag{4}$$

$$\mathbf{c}=\sum_{i=1}^{N} a_i j(x_i). \tag{5}$$

The squared distance between the center $\begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}$ and any point $\begin{bmatrix} j(x_l) \\ d_l \end{bmatrix}$

$$\left\| j(x_l)-\mathbf{c} \right\|^2 + d_l^2 = \left\|\mathbf{c}\right\|^2 - 2(\mathbf{K}\boldsymbol{\alpha})_l + k_{ll} + d_l^2. \tag{6}$$

which does not depend explicitly on the feature map j.

Because of the constraint $\boldsymbol{\alpha}^T\mathbf{1}=1$ in (2), an arbitrary multiple of $\boldsymbol{\alpha}^T\mathbf{1}$ can be added to the objective without affecting its solution. In other words, for an arbitrary $h\in$, (2) yields the same optimal as

$$\begin{aligned} &\max \quad \boldsymbol{\alpha}^T(diag(\mathbf{K})+\boldsymbol{\Delta}-h\mathbf{1})-\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} \\ &s.t. \quad \boldsymbol{\alpha}^T\mathbf{1}=1, \quad \boldsymbol{\alpha}\ge\mathbf{0}. \end{aligned} \tag{7}$$

Hence, any QP problem of the form (7), with the condition (3), can also be regarded as a special MEB problem, called center constrained MEB, i.e. CC-MEB. As pointed out by Tsang et al., CC-MEB can be approximately solved with the asymptotic linear time complexity O(N) and its space complexity independent of N for large datasets by using the generalized core vector machine.

2.1.2. The GCVM Algorithm

The GCVM algorithm is shown in Algorithm 1. Here, the core set, the ball's center, and radius at the *t*th iteration are denoted by $S_t,\mathbf{c}_t$, and R_t respectively. The GCVM algorithm requires the input of a termination parameter e.

The core set can be obtained by using CC-CVM.

Algorithm 1. GCVM

Step 1 Initialize e, $t=0,S_t,\mathbf{c}_t,R_t$

Step 2 Update the core set: if there is no training pattern that falls outside the ball $B(\mathbf{c}_t,(1+e)R_t)$ in the corresponding feature space, $S=S_t$.

Step 3 Find $\mathbf{z}$ such that it is the farthest away from $\mathbf{c}_t$ in the corresponding feature space and set $S_{t+1}=S_t\ \mathbf{U}\{\mathbf{z}\}$

Step 4 Find the new MEB: $B(\mathbf{c}_{t+1},R_{t+1})$

Step 5 Set $t=t+1$, and go to step 2.

2.2. The Weighted Support Vector Machines (WSVM)

2.2.1. Setting the Weight for Imbalanced Problem

To deal with imbalanced dataset, we simply set the weight according to the size of positive and negative dataset. The data in the majority class have to receive lower weight than those in the minority class receives. When the size of positive set is N_{pos} and that of negative set is N_{neg}, the weights are defined as

$$s_i = \begin{cases} 1/N_{pos} & if\ y_i = 1, \\ 1/N_{neg} & otherwise. \end{cases} \quad (8)$$

To maintain the weight ratio and make the convergence speed faster, we also use the following weighting formula

$$s_i = \begin{cases} 1 & if\ y_i = 1,\ N_{pos} \ge N_{neg}, \\ N_{neg}/N_{pos} & if\ y_i = 1,\ N_{pos} < N_{neg}, \\ N_{pos}/N_{neg} & if\ y_i = -1,\ N_{pos} \ge N_{neg}, \\ 1 & if\ y_i = -1,\ N_{pos} < N_{neg}. \end{cases} \quad (9)$$

2.2.2. WSVM

We are given a training set $\{(x_i, y_i)\}_{i=1}^N$ with $y_i \in \{1, -1\}$ .The primal of the weighted SVM (WSVM) is defined as

$$\begin{aligned} \min_{w,b,x} \quad & \frac{1}{2}\|w\|^2 + c\sum_{i=1}^N s_i x_i \\ s.t. \quad & y_i((w \cdot x_i) + b) \ge 1 - x_i, \quad i = 1, \cdots, N, \\ & x_i \ge 0, \quad i = 1, \cdots, N. \end{aligned} \quad (10)$$

By assigning the appropriate weight $s_i, i = 1, \cdots, N$ to each data point, this outperforms the standard SVM with imbalanced training set. The lagrangian of formulations (10) is formulated as

$$\begin{aligned} L = \frac{1}{2}\|w\|^2 + c\sum_{i=1}^N s_i x_i - \sum_{i=1}^N a_i [y_i((w \cdot x_i) + b) - 1 + x_i] \\ - \sum_{i-1}^N b_i x_i \end{aligned} \quad (11)$$

Using the multiplier $a_i \ge 0, b_i \ge 0, i = 1, \cdots, N$. Zeroing the derivative L of with respect to the primal variables yields

$$\begin{cases} \dfrac{\partial L}{\partial w} = w - \sum_{i=1}^N a_i y_i x_i = 0 \\ \dfrac{\partial L}{\partial b} = \sum_{i=1}^N a_i y_i = 0 \\ \dfrac{\partial L}{\partial x_i} = cs_i - a_i - b_i = 0, i = 1, \cdots, N \end{cases} \quad (12)$$

By substituting the primal variables in (11) with the (12), we obtain the dual formulation of (10) as

$$\begin{aligned} \min_a \quad & \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N y_i y_j a_i a_j x_i x_j - \sum_{i=1}^N a_i \\ s.t. \quad & 0 \le a_i \le cs_i, i = 1, \cdots, N. \end{aligned} \quad (13)$$

We have the following decision function

$$f(x) = \mathrm{sgn}(b + \sum_{i=1}^N y_i a_i k(x_i, x)). \quad (14)$$

2.3. The CSWSVM Algorithm

We can now give a fast training algorithm for large datasets which is called the core set weighted support vector machines (CSWSVM). It consists of two stages. The first stage is to obtain the core set of the large training dataset by using CVM. In the second stage, we define the weight on the the obtained core set, WSVM algorithm is utilized to train on the obtained core set and yields a decision function for classifying testing patterns. CSWSVM can be summarized as follows:

Algorithm 2. CSWSVM

Stage 1: Using CVM to obtain the core set.

Step 1 Initialize e , $t = 0, S_t, \mathbf{c}_t, R_t$

Step 2 Update the core set: if there is no training pattern that falls outside the ball $B(\mathbf{c}_t, (1+e)R_t)$ in the corresponding feature space, $S = S_t$.

Step 3 Find $\mathbf{z}$ such that it is the farthest away from $\mathbf{c}_t$, set $S_{t+1} = S_t \mathbf{U} \{\mathbf{z}\}$

Step 4 Find the new MEB: $B(\mathbf{c}_{t+1}, R_{t+1})$

Step 5 Set $t = t + 1$, and go back to second step.

Stage 2: Using WSVM to train the core set S_t .

Step 6 Train the core set using the WSVM algorithm.

Step 7 Yield the decision function according Eq. (14).

3. Experimental Results

In this section, we conduct the performance comparison of the three methods for real problems: Digit, DNA, Letter, Sat, Shuttle, Spambase, Usps, Skin_ Segmentation and MiniBooNE_PID. Most of the datasets are taken from the UCI machine learning repository [17]. Usps is taken from database [18]. All the simulations are carried out in MATLAB7.1 environment running in Intel Core(TM) i5-2400, 3.10GHz, 8GBRAM. The description of datasets is shown in Table 1.

The Gaussian function is taken as the kernel function $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / h)$. where h is the kernel parameter of the Gaussian kernel. The width parameter h is selected to the mean squared norm of the training data, $h = (1/N^2)\sum_{i,j=1}^N \|x_i - x_j\|^2$. We select the approximation parameter e =1e-3. In WSVM, the parameter C was selected from the grid {1, 10, 20, 50, 100, 500, 1000, 10000}, and in GCVM, the parameter v was selected from the grid {1, 10, 20, 50, 70, 100}. The parameter

C 、 v are shown in Table 2. CSWSVM1 and CSWSVM2 are the proposed method with different weight in this paper. Considering the imbalanced nature of the training datasets, the geometric mean accuracy is adopted to evaluate the performance of our algorithms,

$$g = \sqrt{a^+ \cdot a^-}$$

where

$$a^+ = \frac{\#\text{positive samples correctly classified}}{\#\text{ total positive samples classified}} \times 100\%,$$

$$a^- = \frac{\#\text{negative samples correctly classified}}{\#\text{ total negative samples classified}} \times 100\%.$$

This measure has been widely used in dealing with imbalance datasets, and it takes into consideration the classification results on both positive and negative classes.

Ten trials were conducted for the three algorithms and the average results are shown in Tables 3 and 4. Table 3 shows the performance comparison of accuracy of the three methods in the real-world problems; testing accuracy and geometry accuracy of CSWSVM1 and CSWSVM2 is slightly higher than CVM method in most datasets.

Table 4 shows the performance comparison of average training and testing time of the three methods in the real-world problems. As observed from the Table 4, CSWSVM1 and CSWSVM2 methods compare to CVM method with almost same learning speed in most datasets.

Tables 1. Specification of the Datasets.

Datasets	# attributes	# Training (# pos / neg)	# Testing (# pos / neg)
Digit	64	2810 (288 /2522)	2810 (283 / 2527)
DNA	180	3457 (827 /2630)	1729 (404 / 1325)
Letter	16	10000 (391/9609)	10000 (398/9602)
Sat	36	3217 (766/ 2451)	3218 (767/ 2451)
Shuttle	9	29000 (22778 / 6222)	29000 (22808 / 6192)
Spambase	57	2300 (891 / 1409)	2301 (922 / 1379)
Usps	256	6198 (1051/ 5147)	3100 (502/ 2598)
Skin_ Segmentation	3	147034 (30810/116224)	98023 (20049/77974)
MiniBooNE_ PID	50	78038 (21859/56179)	52026 (14640/37386)

Tables 2. Parameters in the Experiment.

Datasets	CVM	CSWSVM1	CSWSVM2
	C		v
Digit	10		2
DNA	1		10
Letter	20		50
Sat	100		10
Shuttle	50		70
Spambase	10000		20
Usps	1000		20
Skin_ Segmentation	10		50
MiniBooNE_ PID	100		10

Tables 3. Comparison of Accuracy of the Three Methods.

Datasets	CVM testing geometry	CSWSVM1 testing geometry	CSWSVM2 testing geometry
Digit	96.4673 98.5632	97.7854 99.0811	99.2333 99.4785
DNA	88.4561 86.8447	94.5618 90.5478	97.3581 96.8862
Letter	93.4698 90.1264	97.7653 96.7814	99.6713 99.0457
Sat	955628 97.7646	96.6742 96.6541	97.8217 97.74
Shuttle	86.0467 84.3741	89.5619 85.7534	90.5481 88.9176
Spambase	77.8128 72.237	73.4671 72.8129	80.5681 78.8641
Usps	98.6291 97.5113	98.8294 98.2242	99.4327 99.7701
Skin_ Segmentation	99.3901 98.3175	99.3711 98.4081	99.4188 99.0118
MiniBooNE_ PID	73.0937 73.3928	73.3665 72.225	79.7522 75.6618

Tables 4. Comparison of Training Time of the Three Methods.

Datasets	CVM	CSWSVM1	CSWSVM2
Digit	4.4461	4.9655	5.0612
DNA	54.4178	58.6559	56.8417
Letter	3.0668	4.5671	4.9035
Sat	3.3379	3.7449	4.1006
Shuttle	1.276	1.9774	2.2109
Spambase	0.2245	0.2312	0.3551
Usps	9.3419	10.7732	9.8775
Skin_ Segmentation	2.2318	3.4377	3.668
MiniBooNE_ PID	1.8475	2.4603	2.4755

4. Conclusions

The CVM utilizes an approximation algorithm for the minimum enclosing ball (CC-MEB) problem. We proposed the core set weighted support vector machines (CSWSVM) approach. It consists of two stages. In the first stage, the core set can be obtained efficiently by using the CVM algorithm. For the second stage, the weighted support vector machine (WSVM) can be used to implement classification. Experiments show that the proposed CSWSVM has comparable performance with CVM implementations.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61170040), by the Natural Science Foundation of Hebei Province (F2011201063 and F2012201023), and by the Key Scientific Research Foundation of Education Department of Hebei Province (ZD2010139).

References

[1] C. Cortes, and V. N. Vapnik, "Support vector networks. Machine Learning," 1995, 20(3), 273–297.

[2] P. Y. Hao, "New support vector algorithms with parametric insensitive/margin model," Neural Networks, 2010, 23(1), 60–73.

[3] M. J. T. David,, and P. W. D. Robert, "Support vector data description. Machine Learning," 2004, 54(1), 45–66.

[4] G. A. Babich, and O. I. Camps, "Weighted Parzen windows for pattern classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, 18(5), 567–570.

[5] N. Takahashi, and T. Nishi, "Rigorous proof of termination of SMO algorithm for support vector machines," IEEE Transactions on Neural Networks, 2005, 16(3), 774–776.

[6] J. Platt, B. Schölkopf, C. Burges, and A. Smola, "Fast training of support vector machines using sequential minimal optimization," In Advances in Kernel Methods – Support Vector Learning. Cambridge, MA: MIT Press, 1999, pp. 185–208.

[7] D. Achlioptas,, F. McSherry,, and B. Schölkopf, "Sampling techniques for kernel methods," In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.), Advances in neural information processing systems, Cambridge, MA: MIT Press, 2002, vol.14, pp. 335–342.

[8] S. Fine, K. Scheinberg, "Efficient SVM training using low-rank kernel representations," Journal of Machine Learning Research, 2001, 2, 243–264.

[9] I. W. Tsang, J. T. Kwok, and J. M. Zurada, "Generalized core vector machines," IEEE Transactions on Neural Networks, 2006, 17(5), 1126-1140.

[10] I. W. Tsang, J. T. Kwok, and P. M. Cheung, "Core vector machines: fast SVM training on very large data sets," Journal of Machine Learning Research, 2005, 6, 363-392.

[11] W. J. Hu, F. L. Chung, S. T. Wang, "The Maximum Vector Angular Margin Classifier and its fast training on large datasets using a core vector machine," Neural Networks, 2012, 27, 60-73.

[12] F. L. Chung, Z. H. Deng, and S. T. Wang, "From minimum enclosing ball to fast fuzzy inference system training on large datasets," IEEE Transactions on Fuzzy Systems, 2009, 17(1), 173–184.

[13] Wenjun Hu, Fu-lai Chung, Shitong Wang, Wenhao Ying, "Scaling up minimum enclosing ball with total soft margin for training on large datasets," Neural Networks, 2012, 36, 120–128

[14] H. He, E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng. 2009, 21 (9), 1263–1284.

[15] Y. M. Sun, K. C. Wong, and M.S. Kamel, "Classification of imbalanced data: a review," International journal of pattern recognition and artificial intelligence, 2009, 23(4), 687-719.

[16] Weiwei Zong, GB Huang, Yiqiang Chen, "Weighted extreme learning machine for imbalance learning," Neurocomputing, 2013, 101, 229-242.

[17] A. Frank, A. Asuncion, UCI machine learning repository, 2010. URL http: //archive.ics.uci.edu/ml.

[18] J. J. Hull, "A database for handwritten text recognition research," IEEE Trans. Pattern Anal. Mach. Intell., 1994, 16 (5), 550-554.