Feature selected cost-sensitive twin SVM for imbalanced data

Xiaopeng Li^{1,*}, and Xianrong Zhang²

¹Zhejiang normal university, jinhua city, zhejiang province, China

²Keyuan new village, jinzhai road, baohe district, hefei city, anhui province, China

Keyword: Twin SVM, cost-sensitive, feature selection, imbalanced data.

Abstract. In this paper, we propose a cost-sensitive twin SVM (cs-tsvm) and apply it to imbalanced data. A weight is added to each instance according to its cost of misclassification which is related to its position. In preprocessing part, features are selected by their difference of majority and minority classes. The feature is selected when its difference value is higher than average one. The experiment is conducted on UCI datasets and G-mean, AUC and accuracy are evaluation metrics. The experimental results show that Feature selection with CS-TWSVM is useful for datasets with high dimension.

1 Twin support vector machine

In 2007, Twin Support Vector Machine (TWSVM) is proposed by Jayadeva et al [1,2,3]. It obtains nonparallel planes around which the data points of the corresponding class get clustered. Its nonparallel planes are obtained by solving the following pair of quadratic programming problem.

$$(TWSVM1) \min_{w^{(1)}, b^{(1)}, q} \frac{1}{2} (Aw^{(1)} + e_1 b^{(1)})^T (Aw^{(1)} + e_1 b^{(1)}) + c_1 e_2^T q \quad (1)$$

$$s. t. -(Bw^{(1)} + e_2 b^{(1)}) + q \geq e_2, q \geq 0 \quad (2)$$

$$(TWSVM2) \min_{w^{(2)}, b^{(2)}, q} \frac{1}{2} (Bw^{(2)} + e_2 b^{(2)})^T (Bw^{(2)} + e_2 b^{(2)}) + c_2 e_1^T q \quad (3)$$

$$s. t. -(Aw^{(2)} + e_1 b^{(2)}) + q \geq e_1, q \geq 0 \quad (4)$$

where A, B are instances in two classes separately, $c_1, c_2 > 0$ are parameters and e_1 and e_2 are vectors of ones of appropriate dimensions.

The algorithm finds two hyperplanes, one for each class, and classifies points according to which hyperplane a given point is closest to [4,5,6]. The first term in the objective function of (1) or (2) is the sum of squared distances from the hyperplane to points of one class.

* Corresponding author: 13065153072@163.com

Therefore, minimizing it tends to keep the hyperplane close to points of one class. The constraints require the hyperplane to be at a distance of at least 1 from points of the other class; a set of error variables is used to measure the error wherever the hyperplane is closer than this minimum distance of 1. The second term of the objective function minimizes the sum of error variables, thus attempting to minimize misclassification.

Original TWSVM can be utilized in imbalanced data. We can assign different values to c_1, c_2 . The cost of misclassifying an instance in minority class to majority class is higher than the cost of misclassifying an instance in majority class to minority class [7,8].

As for error variant q , hinge loss function is used to calculate q . Hinge loss function is defined as follows:

$$L[y(w \cdot x + b)] = [1 - y(w \cdot x + b)]_+ \quad (5)$$

In this paper, we introduce a new algorithm in section 2 by eliminating the second term in TWSVM and add weight of each instance to first term. In experiment part, we compare the performance of these two classifiers in imbalanced datasets using G-mean and AUC as metrics.

2 CS-TWSVM

2.1 Feature Selection

The feature is selected if its feature deviation value is higher than average. Feature deviation is the deviation between the average value of the feature in majority class and the average value of it in minority class. The FD (Feature Deviation) [9] is defined as follows:

Training set: $T(x_1, y_1), \dots, (x_m, y_m), (x_{m1}, y_{m1}), \dots, (x_{mk}, y_{mk})$
 $(A_1, A_1), \dots, (A_m, 1), (B_1, -1), \dots, (B_k, -1)$,
 where $A_i(a_{i1}, \dots, a_{in}) \ T \in R_n \ (i = 1, \dots, m), \ B_j(b_{j1}, \dots, b_{jn}) \ T \in R_n \ (j = 1, \dots, k)$
 $x_l \in R_n, y_l \in [-1, 1], (l = 1, \dots, m_k)$

$$FD_j = \frac{|A_j - B_j|}{\max(A_j, B_j)} \quad (6)$$

$$A_j = \frac{\sum_{i=1}^n a_{ij}}{n} \quad (7)$$

$$B_j = \frac{\sum_{i=1}^n b_{ij}}{n} \quad (8)$$

2.2 Instance weights

The weight for each instance in a class is related to the distance between itself to the central of opposite class. The weights of instances in majority class and minority class are defined in different way. The weights of instances in minority class are negatively related to distance between itself and the central of majority class. The closer to the center of majority class the instance is, the higher weight instance has. In the contrary, the weights of instances in majority class are positively related to distance between itself and the central of majority class. The further from the center of minority class the instance is, the higher weight the instance has. The weight of instance j in minority class is defined as follows:

$$\text{weight}_j = \max_i d_i + \min_i d_i - d_j \quad (9)$$

where d_i is the distance between instance i and the center of majority class.

The weight of instance j in majority class is defined as follow:

$$\text{weight}_j = d_j \quad (10)$$

where d_j is the distance between instance j and the center of minority class.

2.3 CS-TWSVM (cost-sensitive twin SVM)

The weight introduced in section 3.2 is added to object function. Our goal of using the weight is to minimize the cost of misclassification by pushing hyperplane of majority class away from minority class and setting hyperplane of minority class near the instance close to majority class. A hypothetic situation is described in figure 1. Figure 1(a) is two nonparallel hyperplane of original TWSVM. Figure 1(b) is two nonparallel hyperplane modified by adding weight [10,11,12].

From Figure 1, we can see that with weight there are less chance of misclassifying instance in minority class.

The object function is defined as follows:

$$\min(w_1, b_1) \frac{1}{2} \|Z_1 X_1 w_1 + e_1 b_1\|^2 \quad (11)$$

$$s. t. -(Z_1 X_2 w_1 + e_2 b_1) \geq e_2 \quad (12)$$

$$\min(w_2, b_2) \frac{1}{2} \|Z_2 X_2 w_2 + e_2 b_2\|^2 \quad (13)$$

$$s. t. -(Z_2 X_1 w_2 + e_1 b_2) \geq e_1 \quad (14)$$

where Z_1, Z_2 are weight matrixes of minority class and majority class. e_1 and e_2 represent two vectors of suitable dimension and having all values as 1's. Equation (1) and (2) are further used to develop CS-TWSVM classifier. Lagrangian of equation (1) is obtained as:

$$L(w_1, b_1, \alpha) = \frac{1}{2} \|Z_1 X_1 w_1 + e_1 b_1\|^2 + \alpha^T (Z_1 X_2 w_1 + e_2 b_1 + e_2) \quad (15)$$

Lagrangian multipliers are denoted by vector α . Karush-kuhn-Tucker(KKT) conditions of equation (3) are formulated as

$$\frac{\partial L}{\partial w_1} = (Z_1 X_1)^T (Z_1 X_1 w_1 + e_1 b_1) + (Z_1 X_2)^T \alpha = 0 \quad (16)$$

$$\frac{\partial L}{\partial b_1} = e_1^T (Z_1 X_1 w_1 + e_1 b_1) + e_2^T \alpha = 0 \quad (17)$$

$$-(Z_1 X_2 w_1 + e_2 b_1) \geq e_2 \quad (18)$$

$$\alpha^T (Z_1 X_2 w_1 + e_2 b_1 + e_2) = 0 \quad (19)$$

$$\alpha \geq 0, \beta \geq 0 \quad (20)$$

Following equation (9) is obtained after merging equation (4), (5):

Let

$$\begin{bmatrix} X_1^T Z_1^T \\ e_1^T \end{bmatrix} [X_1^T Z_1^T \quad e_1^T] \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} X_2^T Z_1^T \\ e_2^T \end{bmatrix} \alpha = 0 \quad (21)$$

$$A = [X_1^T Z_1^T \quad e_1^T] \quad B = [X_2^T Z_1^T \quad e_2^T] \quad u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \quad (22)$$

With these notation, equation (9) is reformulated as:

$$A^T A u_1 + B^T \alpha = 0 \quad (23)$$

$$u_1 = -(A^T A)^{-1} B^T \alpha \quad (24)$$

At times it is difficult to obtain the inverse of ATA . This condition is handled by adding regularization term δI in the above equation, where ' I ' represents an identity matrix of suitable dimension. Equation (17) is reformulated as follows:

$$u_1 = -(A^T A + \delta I)^{-1} B^T \alpha \quad (25)$$

In the same manner, normal vector and bias for the second class are achieved by solving equation as follows:

$$u_2 = -(B^T B + \delta I)^{-1} A^T \alpha \quad (26)$$

Normal vectors and biases are further used to generate nonparallel planes in equations below:

$$x^T w_1 + b_1 = 0 \quad \text{and} \quad x^T w_2 + b_2 = 0 \quad (27)$$

In this way, CS_TWSVM determines hyper-plane for each class and a new data sample is assigned to a class by using following decision function:

$$Class_i = \min |x^T w_i + b_i| \quad f \quad \text{or} \quad i = 1, 2 \quad (28)$$

The perpendicular distance of the test data sample is calculated from each hyper-plane and pattern is assigned to the class from which its distance is lesser.

3 Experiment

3.1 Feature Selection

We use FD value of each feature between positive and negative class formatted by (1) as a metric for feature selection[14,15]. The features with higher FD value than average one are selected. For high dimensional datasets, the advantage of feature selection is very obvious. Especially for some practical applications with from tens to hundreds of variables or features, feature selection can reduce computation time, reduce the effect of curse of dimensionality, and improve the prediction performance. The performance of CS-TWSVM classifier with feature selection and without it is compared in experiment 3.3. Cont.Table1

Table 1. Feature selection of each dataset.

Dataset name	Ionosphere	Hepatitis	Pima Diabetes	India
Feature set	[9,14,20,21,22,28,32,33]	[0,14,15,16,19]	[0,8]	
Dataset name	Sonar			Heart Statlog
Feature set	[0,1,2,3,4,7,8,9,10,11,12,33,34,35,36,42,43,44,45,46,47,48,49,50,51,53,57,60]			[0,8,9,11,13]

3.2 The performance of CS-TWSVM

The performance of CS-TWSVM is compared with SVM, TSVM, Least Square TSVM and Weighted LS-TSVM. SVM, TSVM, and Least Square TSVM are not specific for imbalance data, while weighted LS-TSVM is a refinement of Least Square TSVM for imbalance data.

Table 2. Performance Comparison of different Classifiers using Gaussian Kernel Function.

Dataset	Performance Measurement Parameters	SVM	TSVM	Least Square TSVM	Weighted LS-TSVM	CS-TWSVM
Ionosphere	Accuracy	0.8604	0.8803	0.8993	0.8824	0.8919
	Sensitivity	0.8823	0.7169	0.7495	0.7267	0.9200
	Specificity	0.7934	0.9827	0.9576	1.0000	0.8400
	Geometric Mean	0.8366	0.8393	0.8473	0.8524	0.8756
Hepatitis	Accuracy	0.8213	0.8167	0.8428	0.8667	0.8750
	Sensitivity	0.8505	0.8495	0.8017	0.9167	0.8667
	Specificity	0.6729	0.6703	0.8723	0.6667	0.9000
	Geometric Mean	0.7565	0.7544	0.8362	0.8913	0.8832
Pima Indian Diabetes	Accuracy	0.7589	0.7428	0.7761	0.7567	0.7800
	Sensitivity	0.5176	1.0000	0.8500	0.8709	0.7667
	Specificity	0.8875	0.2523	0.7103	0.6957	0.8000
	Geometric Mean	0.6778	0.5022	0.7770	0.7783	0.7831

From Table 2, we can see that the performance of CS-TWSVM is better than other classifiers in most datasets. G-mean is more important metric to estimate the performance of classifier for imbalanced datasets.

3.3 Performance comparison

Table 3. Comparison between feature selection and without it.

Dataset	Ionosphere		Hepatitis		Pima Indian Diabetes		Sonar		HeartStatlog	
	Without FS	FS	Without FS	FS	Without FS	FS	Without FS	FS	Without FS	FS
Accuracy	0.892	0.932	0.875	0.875	0.780	0.740	0.870	0.900	0.875	0.875
Sensitivity	0.920	0.946	0.867	0.867	0.767	0.667	0.875	0.909	1.000	1.000
Specificity	0.840	0.900	0.900	0.900	0.800	0.800	0.867	0.900	0.849	0.849
Geometric Mean	0.876	0.921	0.883	0.883	0.783	0.871	0.871	0.905	0.921	0.921

4 Conclusion

From Table 3, the performance of CS-TWSVM with feature selection is obviously much better than CS-TWSVM without it in Ionosphere and Sonar datasets. But in Hepatitis and Heart-Statlog datasets, the performance is the same and even in Pima India Diabetes, the

performance with feature selection is worse than without it. This is because Ionosphere and Sonar are high dimensional datasets, while Hepatitis, Heart-Statlog, and Pima India Diabetes are not, especially, Pima Indian Diabetes only has eight features. Feature selection is more useful for datasets with high dimension. Feature selection is more useful for datasets with high dimension.

References

1. Khedher L, Ram rez J, Go rriz JM, Brahim A, Segovia F. Alzheimer's disease neuroimaging initiative. Early diagnosis of Alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented MRI images. *Neurocomputing* 2014.
2. Xie J, Wang C. Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato- squamous diseases. *Expert Syst Appl* 2011;38(5):5809–15.
3. Wang TY, Chiang HM. One-against-one fuzzy support vector machine classifier: an approach to text categorization. *Expert Syst Appl* 2009;36(6):10030–4.
4. Wang TY, Chiang HM. Solving multi-label text categorization problem using support vector machine approach with member- ship function. *Neurocomputing* 2011;74(17):3682–9.
5. Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K. An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Syst Appl* 2012;39(1):424–30.
6. Jayadeva, Khemchandani R, Chandra S. Twin support vector machine for pattern classification. *IEEE Trans Pattern Anal Mach Intell* 2007;29(5):905–10.
7. Shao YH, Zhang CH, Wang XB, Deng NY. Improvements on twin support vector machines. *IEEE Trans Neural Netw* 2011;22(6).
8. Peng X. A v-twin support vector machine classifier and its geometric algorithms. *Inf Sci* 2010;180:3863–75.
9. Kumar MA, Gopal M. Least squares twin support vector machines for pattern classification. *Expert Syst Appl* 2009;36:7535–43.
10. Xu Y, Guo R, Wang L. A twin multi-class classification support vector machine. *Cogn Comput* 2013;5(4):580–8.
11. Peng X. TSVR: an efficient twin support vector machine for regression. *Neural Netw* 2010;23:365–72.
12. Peng X, Xu D, Shen J. A twin projection support vector machine for data regression. *Neurocomputing* 2013;138:131–41.
13. N. V. Chawla, K. Bowyer, L. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of AI Research*, vol. 16, pp. 321–357, 2002.
14. S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE— majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.
15. Peng Cao, Dazhe Zhao, and Osmar Zaiane, An Optimized Cost-Sensitive SVM for Imbalanced Data Learning. *PAKDD 2013, Part II, LNAI 7819*, pp. 280–292, 2013.
16. J. L. Hsu, P. C. Hung, and H. Y. Lin, "Applying under- sampling techniques and cost-sensitive learning methods on risk assessment of breast cancer," *Journal of Medical Systems*, vol. 39, no. 4, pp. 1–13, 2015.