International Journal of Database Theory and Application
Vol.7, No.2 (2014), pp.25-36
http://dx.doi.org/10.14257/ijdta.2014.7.2.03

Weighted Least Square Twin Support Vector Machine for Imbalanced Dataset

Divya Tomar, Shubham Singhal and Sonali Agarwal

Indian Institute of Information Technology, Allahabad, India
divyatomar26@gmail.com, shubham07iiit@gmail.com and sonali@iiita.ac.in

Abstract

This research work proposes a Weighted Least Square Twin Support Vector Machine (WLSTSVM) for imbalanced dataset. Real world data are imbalanced in nature due to which most of the classification techniques do not work well. In Imbalanced data, there is a huge difference between the numbers of data samples of classes. One class data samples are larger as compared to other class data samples. This paper discusses the traditional methods of handling imbalanced data and proposes an improvement over Least Square Twin Support Vector Machine. This research work has performed experiment on five benchmark UCI datasets using 10-fold cross validation method. The results of experiment show that the proposed technique performed well for imbalanced dataset and its accuracy is better as compared to other existing methods. This research work presents the formulation of proposed approach for both linear and non-linear data samples.

Keywords: *Weighted Least Square Twin Support Vector Machine, Twin Support Vector Machine, Support Vector Machine, Imbalanced data*

1. Introduction

Dataset are said to be imbalanced if the data samples belonging to one class outnumbers the data samples belonging to the other class i.e., one class has lesser number of data samples as compared to the other class. In this paper, the class with more and less number of data samples is referred as ‘majority’ and ‘minority’ class respectively. In many real world applications, for example disease detection, fraudulent telephone calls and image and text classification, samples of different classes may be distributed non-uniformly to the classes. Over the years, the imbalanced dataset problem has been noticed frequently and has become a prime concern because the performance of algorithms significantly degrades due to it. So, to enhance classifier's performance, it is necessary to handle the imbalanced dataset problem. There are mainly two approaches to deal with this issues- either adjusting the distribution of data or adjusting the classifier to adapt them to imbalanced dataset. The simplest method to handle data imbalance problem is to balance the dataset. There exist two methods for balancing the dataset: under-sampling and over-sampling. Under-sampling method reduces the data samples of majority class to balance it with the data samples of minority class. This method is simple but it loses the information of a class with more data samples [1-5]. While the over-sampling method duplicates or increases the number of data samples of class with less data samples till both classes have same number of data samples. Experimental analysis highlights the better performance of under-sampling method as compared to over-sampling method. The reason behind this is in over-sampling method there is no loss of information of majority class but unnatural bias is generated in support of minority class [4]. The second method to deal with class imbalanced is to adjust the classifier which is more appropriate than

ISSN: 2005-4270 IJDTA

balancing the dataset. Assigning different cost to training samples seems to be an efficient approach where data samples could be assigned with different costs [6-10]. Various approaches are given by researchers to deal with data imbalanced problem [11-15]. Akbani et al. have applied Support Vector Machine (SVM) to handle this issue [11]. This research work proposed a new approach which was the combination of SMOTE with different error costs to imbalanced dataset. Batista et al. have performed an analysis of the behavior of several methods which are used to balance the machine learning training data [12]. Different ensemble approaches such as boosting SVM and random forest are also suggested by researchers to deal with class imbalance problem [13-15]. Several weighted approaches to handle the class imbalanced problem are proposed by many researchers [16-19]. Yang et al. and Suykens proposed a weighted SVM and LSSVM approach respectively [16, 17]. Fung et al. introduced weight factor in the formulation of proximal SVM to deal with imbalanced data samples for multiclass problem [18]. Weighted approach for Least Square Twin Support Vector Machine also proposed by Chen et al.[19]. All these research have found that classification approach based on weighted approach improves the prediction performance of a classifier. It is also found that modified SVM has great capability to solve the class imbalance problem and has great predictive capability as compared to other existing methods. Vapnik et al. proposed SVM in 1990s which is based on the statistical learning theory and follows Structural Risk Minimization principle (SRM) [20, 21]. SVM works well for both binary as well multi class problems [22-24]. It separates the data points of different classes by constructing hyper-planes that maximizes the separation between them. If the data are non-linearly separable then SVM maps the data points in high dimensional space using kernel function. In high dimension space, it constructs kernel surfaces to separate data points [25]. Thus in this way it is suitable for both linearly separable and non-linearly separable data points. As compared to other machine learning approaches, SVM has several advantages. Firstly, for data classification it obtains unique global solution rather than local. Secondly, it maximizes the generalization capability by minimizing the training error of data samples. Thus along with these advantages, researchers face various challenges with SVM. One challenge with SVM is its complexity because Quadratic Programming Problem (QPP) is difficult to solve. If ‘l’ indicates the size of training data samples then its computational complexity is $O(l^3)$, which is very expensive. Jayadeva et al. proposed a novel Twin Support Vector Machine, a solution of binary classification to enhance the computational speed [26, 27]. Rather than solving one complex QPP, TSVM solves two smaller QPPs and determines two non-parallel hyper-planes. The size of quadratic problem in TSVM is (l/2) and its computational complexity is $O\ ((^{l}/_{2})^{3})$. So, TSVM reduces the computational complexity by the factor of 4 *i.e.*, it is 4 times faster than to traditional SVM [26-28].

Least Square Twin Support Vector Machine (LSTSVM), proposed by Kumar et al., has some enhanced features such as better generalization capability and lesser computational time as compared to standard TSVM [29]. LSTSVM associated with additional weight parameters is introduced in this research work which indicates significant improvement in the performance of minority class without having any influence on the classification of majority class. This research paper proposed a Weighted LSTSVM (WLSTSVM) approach to handle imbalance data problem of binary class by assigning distinct weight to training samples. The experiment has been performed on five benchmark UCI datasets using 10-fold cross validation method.

The paper is organized in 5 sections. The Section 2 briefly explains the formulation of Least Square Twin Support Vector Machine. In Section 3, the proposed approach to handle class imbalance problem is discussed. The results of experiment on five real datasets from

UCI and performance estimators of proposed approach are discussed in Section 4 and finally the Section 5 contains the conclusion of this research work.

2. Least Square Twin Support Vector Machine

2.1. For Linearly Separable Data

Least Square TSVM (LSTSVM) was proposed by Kumar et al. to speed up the classification process [29]. It solves two linear equations rather than two complex quadratic programming problems. LSTSVM is faster than TSVM and has better generalization capability. Consider 'm' data points belong to positive class and 'n' data points belong to negative class. R represents training sample of 'k' dimension. Let $\boldsymbol{X_1} \in \boldsymbol{R^{m\times k}}$ is a matrix that represents the instances of positive class and $\boldsymbol{X_2} \in \boldsymbol{R^{n\times k}}$ is a matrix that represents the instances of negative class [28]. Two non-parallel hyper-planes in $\boldsymbol{R^k}$ are given below:

$$x^T w_1 + b_1 = 0 \text{ and } \qquad x^T w_2 + b_2 = 0 \tag{1}$$

The primal problem of LS-TSVM is as follows:

$$\min(w_1, b_1, \xi) \qquad \frac{1}{2}\|X_1 w_1 + e_1 b_1\|^2 + \frac{c_1}{2}\xi^T\xi$$

$$\text{s.t.} - (X_2 w_1 + e_2 b_1) + \xi = e_2 \tag{2}$$

and

$$\min(w_2, b_2, \eta) \qquad \frac{1}{2}\|X_2 w_2 + e_2 b_2\|^2 + \frac{c_2}{2}\eta^T\eta$$

$$\text{s.t. } (X_1 w_2 + e_1 b_2) + \eta = e_1 \tag{3}$$

where $'\xi'$and $'\eta'$ are slack variable whereas e1 and e2 are the two vectors having all values as 1s with suitable dimensions. 'c1' and 'c2' are penalty parameters. Lagrangian of the above equation is given as:

$$L(w_1, b_1, \xi, \alpha) = \frac{1}{2}\|X_1 w_1 + e_1 b_1\|^2 + \frac{c_1}{2}\xi^T\xi - \alpha^T(-(X_2 w_1 + e_2 b_1) + \xi - e_2) \tag{4}$$

$$L(w_2, b_2, \eta, \beta) = \frac{1}{2}\|X_2 w_2 + e_2 b_2\|^2 + \frac{c_2}{2}\eta^T\eta - \beta^T((X_1 w_2 + e_1 b_2) + \eta - e_1) \tag{5}$$

Where $\alpha \in R^n$ and $\beta \in R^n$ are the vectors of Lagrangian multiplier. KKT condition of above equation:

$$\frac{\partial L}{\partial w_1} = X_1^T(X_1 w_1 + e_1 b_1) + X_2^T\alpha = 0 \tag{6}$$

$$\frac{\partial L}{\partial b_1} = e_1^T(X_1 w_1 + e_1 b_1) + e_2^T\alpha = 0 \tag{7}$$

$$\frac{\partial L}{\partial \xi} = c_1\xi - \alpha = 0 \tag{8}$$

$$\frac{\partial L}{\partial \alpha} = -(X_2 w_1 + e_2 b_1) + \xi - e_2 = 0 \tag{9}$$

Equation 6 and 7 when combined gives the following equation:

$$\begin{bmatrix} X_1^T \\ e_1^T \end{bmatrix} [X_1 \ \ e_1] \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} X_2^T \\ e_2^T \end{bmatrix} \alpha = 0 \tag{10}$$

Let H= $[X_1 \ \ e_1]$ and G=$[X_2 \ \ e_2]$. Using equation 8, 9 and 10 the following values are obtained:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(G^T G + \frac{1}{c_1} H^T H\right)^{-1} G^T e_2 \tag{11}$$

And

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(H^T H + \frac{1}{c_2} G^T G\right)^{-1} H^T e_1 \tag{12}$$

After solving above two equations, we obtain weights and biases for the calculation of two non-parallel hyper-planes by using equation 1 and assign new data into a class as follows:

$$\text{class i} = \arg\min(j = 1,2) \frac{|w_j^T x + b_j|}{\|w_j\|} \tag{13}$$

2.2. For non-linear separable Data

LSTSVM is also useful for the classification of non-linearly separable data samples. Kernel functions such as Polynomial, Gaussian etc. are used for this purpose. LSTSVM for non-linear data samples is obtained as follows:

$$\min(\mu_1, \gamma_1, \xi) \qquad \frac{1}{2}\left\|K(X_1, D^T)\mu_1 + e\gamma_1\right\|^2 + \frac{c_1}{2}\xi^T \xi$$

$$\text{s.t.} \qquad -\left(K(X_2, D^T)\mu_1 + e\gamma_1\right) = e - \xi \tag{14}$$

and

$$\min(\mu_2, \gamma_2, \xi) \qquad \frac{1}{2}\left\|K(X_2, D^T)\mu_2 + e\gamma_2\right\|^2 + \frac{c_2}{2}\eta^T \eta$$

$$\text{s.t.} \qquad \left(K(X_1, D^T)\mu_2 + e\gamma_2\right) = e - \eta \tag{15}$$

where D=$[X_1 \ \ X_2]^T$. After substituting P= $[K(X_1, D^T) \ \ e]$, Q= $[K(X_2, D^T) \ \ e]$, We can further achieve :

$$\begin{bmatrix} \mu_1 \\ \gamma_1 \end{bmatrix} = -(Q^T Q + \frac{1}{c_1} P^T P)^{-1} Q^T e \tag{16}$$

$$\begin{bmatrix} \mu_2 \\ \gamma_2 \end{bmatrix} = (P^T P + \frac{1}{c_2} Q^T Q)^{-1} P^T e \tag{17}$$

Following are the kernel generated surfaces instead of planes:

$$K(x^T, D^T)\mu_1 + \gamma_1 = 0 \text{ and } K(x^T, D^T)\mu_2 + \gamma_2 = 0 \tag{18}$$

After solving equation 16 and 17, the value of μ_1, μ_2, γ_1 and γ_2 are obtained and further used to generate kernel generated surfaces. The new data point assigned to a particular class is evaluated as follows:

$$\text{class(i)} = \text{argmin}(j = 1,2) \frac{|x^T \mu_j + \gamma_j|}{\|\mu_j\|} \tag{19}$$

For every data point its distance from kernel surfaces is calculated and respectively assigned to a class having lesser distance. x_i and x_j are the vectors in the input space and Gaussian and Polynomial kernel functions are represented as:

$$\text{Gaussin Kernel Function: } K_G = \exp\left(-\frac{\|x_i - x_j\|^2}{2\,\sigma^2}\right) \qquad (20)$$

$$\text{Polynomial Kernel Function: } K_P(x_i, x_j) = (1 + x_i^T x_j)^d \qquad (21)$$

Above stated equations are used to develop a LSTSVM classifier model.

3. Weighted Least Square Twin Support Vector Machine

In this research, the classifier adjustment for solving the class imbalance problem has been focused and performed by using WLSTSVM in which different weights are assigned to different data samples. Therefore, suitable weight selection is an important issue in proposed approach. Appropriate weight selection has to satisfy following two conditions:

a) The higher weight is assigned to the data samples of minority class while the majority class data samples receive lower weight.

b) Weight should lie within 0 to 1 range *i.e.*, 0<weight≤1 so that WLSTSVM could be trained with significant convergence.

The proposed approach assigns a weight to majority and minority class according to their size. Let N_{class1} and N_{class2} indicate the number of data samples of class 1 and class 2 respectively. For this purpose, following weighted formula is used to deal with imbalanced class problem:

For Class 1: if ($N_{class1} \geq N_{class2}$) Weight= 1
else Weight= N_class2/N_class1

For Class 2: if($N_{class1} \geq N_{class2}$) Weight= N_class1/N_class2
else Weight= 1 (22)

3.1. For Linearly Separable Data

Let X1 and X2 are two matrices representing the data samples of positive and negative classes respectively. R is the real space of ‘n’ dimension and V indicates the diagonal matrix containing weight for each data sample which is calculated by using equation 22. For linearly separable data points, the WLSTSVM is given below:

$$\min(w_1, b_1, \xi) \quad \frac{1}{2}\,\|X_1 w_1 + e_1 b_1\|^2 + \frac{c_1}{2}\,\xi^T \sum_{i=1}^{m_2} V_i\,\xi$$

$$\text{s.t.} - (X_2 w_1 + e_2 b_1) + \xi = e_2 \qquad (23)$$

and

$$\min(w_2, b_2, \eta) \quad \frac{1}{2}\,\|X_2 w_2 + e_2 b_2\|^2 + \frac{c_2}{2}\,\eta^T \sum_{j=1}^{m_1} V_j\,\eta$$

$$\text{s.t.} (X_1 w_2 + e_1 b_2) + \eta = e_1 \qquad (24)$$

where c_1 , c_2 are known penalty factors whereas e_1 and e_2 are the two vectors having all values as 1's with suitable dimensions. Instances of positive/negative classes are indicated by m_1 and m_2 correspondingly. The lagrangian of equation (23) and (24) are given below:

$$L(w_1, b_1, \xi, \alpha) = \frac{1}{2} \|X_1 w_1 + e_1 b_1\|^2 + \frac{c_1}{2} \xi^T V_i \xi - \alpha^T(-(X_2 w_1 + e_2 b_1) + \xi - e_2) \tag{25}$$

$$L(w_2, b_2, \eta, \beta) = \frac{1}{2} \|X_2 w_2 + e_2 b_2\|^2 + \frac{c_2}{2} \eta^T V_j \eta - \beta^T((X_1 w_2 + e_1 b_2) + \eta - e_1) \tag{26}$$

Where $\alpha \in R^n$ and $\beta \in R^n$ are the vectors of Lagrangian multiplier. KKT condition of above equation:

$$\frac{\partial L}{\partial w_1} = X_1^T(X_1 w_1 + e_1 b_1) + X_2^T \alpha = 0 \tag{27}$$

$$\frac{\partial L}{\partial b_1} = e_1^T(X_1 w_1 + e_1 b_1) + e_2^T \alpha = 0 \tag{28}$$

$$\frac{\partial L}{\partial \xi} = c_1 V_i \xi - \alpha = 0 \tag{29}$$

$$\frac{\partial L}{\partial \alpha} =-(X_2 w_1 + e_2 b_1) + \xi - e_2 = 0 \tag{30}$$

Combining both the Langrangian equations (25 and 26), the following equation is obtained:

$$\begin{bmatrix} X_1^T \\ e_1^T \end{bmatrix} [X_1 \ \ e_1] \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} X_2^T \\ e_2^T \end{bmatrix} \alpha = 0 \tag{31}$$

Let $u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$, H= [X_1 e_1] and G=[X_2 e_2] (32)

Equation 29 can be rewritten as:

$$H^T H u_1 + G^T \alpha = 0 \tag{33}$$

After solving above equation, we get:

$$u_1 = -(H^T H)^{-1} G^T \alpha \tag{34}$$

Sometime it is not possible to calculate inverse, in that condition,

$$u_1 = -(H^T H + \delta I)^{-1} G^T \alpha \tag{35}$$

Where I is an identity matrix of suitable dimension and $\delta > 0$. After solving equation 29, 30 and 33 we obtain:

$$\alpha = \left[\frac{D_i^{-1}}{c_1} + G(H^T H)^{-1} G^T\right]^{-1} e_2 \tag{36}$$

Where D_i is a diagonal matrix having equivalent weight V_i corresponding to data points of negative class. In same manner,

$$u_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (G^T G)^{-1} H^T \beta \tag{37}$$

and

$$\beta = \left[\frac{D_j^{-1}}{c_2} + H(G^T G)^{-1} H^T\right]^{-1} e_1 \tag{38}$$

Where D_j is a diagonal matrix having equivalent weight V_j corresponding to data points of positive class. Equation 35 and 37 are used to determine two non-parallel hyper-planes:

$$x^T w_1 + b_1 = 0 \text{ and } x^T w_2 + b_2 = 0 \tag{39}$$

The new data point assigned to a particular class is evaluated as follows:

$$\text{Class i= } \min\left|x^T w_i + b_i\right| \text{ for } i = 1{,}2 \tag{40}$$

For every data point its distance from hyper-planes is calculated and respectively assigned to a class having lesser distance

Algorithm:

Step 1. Define H and G matrix by equation 32.
Step 2. Compute weight using equation 22.
Step 3. c1 and c2 are selected on the basis of validation.
Step 4. Determine Lagrangian parameters α and β using equation 36 and 38 and then calculate u_1 and u_2 using equation 34 and 37.
Step 5. Determine two non-parallel line using equation 39.
Step 6. Classify new patterns by using equation 40.

3.2. For non-linear Separable Data

WLSTSVM also provides solution for non-linearly separable data classification with the help of kernel function. Let K represents the kernel function; following are the equations of kernel generated surfaces:

$$K(x^T, D^T)\mu_1 + \gamma_1 = 0 \text{ and } K(x^T, D^T)\mu_2 + \gamma_2 = 0 \tag{41}$$

WLSTSVM for non-linear data samples is obtained as follows:

$$\min(\mu_1, \gamma_1, \xi) \quad \frac{1}{2}\left\|K\left(X_1, D^T\right)\mu_1 + e_1\gamma_1\right\|^2 + \frac{c_1}{2}\xi^T \sum_{i=1}^{m_2} V_i\, \xi$$

$$\text{s.t.} \quad -\left(K\left(X_2, D^T\right)\mu_1 + e_2\gamma_1\right) = e_2 - \xi \tag{42}$$

and

$$\min(\mu_2, \gamma_2, \xi) \quad \frac{1}{2}\left\|K\left(X_2, D^T\right)\mu_2 + e_2\gamma_2\right\|^2 + \frac{c_3}{2}\eta^T \sum_{j=1}^{m_1} V_j\, \eta$$

$$\text{s.t.} \quad \left(K\left(X_1, D^T\right)\mu_2 + e_1\gamma_2\right) = e_1 - \eta \tag{43}$$

where $D=[X_1 \ \ X_2]^T$, After substituting $P= \left[K\left(X_1, D^T\right) \ \ e_1\right]$, $Q= \left[K\left(X_2, D^T\right) \ \ e_2\right]$, we get:

$$z_1 = \begin{bmatrix}\mu_1 \\ \gamma_1\end{bmatrix} = -\left(P^T P\right)^{-1} Q^T \alpha \tag{44}$$

$$\alpha = \left[\frac{{D_i}^{-1}}{c_1} + Q(P^T P)^{-1} Q^T\right]^{-1} e_2 \tag{45}$$

$$z_2 = \begin{bmatrix}\mu_2 \\ \gamma_2\end{bmatrix} = \left(Q^T Q\right)^{-1} P^T \beta \tag{46}$$

$$\beta = \left[\frac{{D_j}^{-1}}{c_2} + P(Q^T Q)^{-1} P^T\right]^{-1} e_2 \tag{47}$$

After solving above equations, the value of μ_1, μ_2, γ_1 and γ_2 are obtained which are further used to generate kernel surfaces and the classes are assigned to data samples by calculating their distances from kernel generated surfaces as follows:

$$\text{Class i= min}\left|K(x^T, D^T)\mu_i + \gamma_i\right| \text{ for } i = 1,2 \tag{48}$$

For every data point its distance from kernel surfaces is calculated and respectively assigned to a class having lesser distance

Algorithm:

Step 1. Define matrix D.
Step 2. Select kernel function for non-linear data samples. Define P and Q matrix.
Step 3. Compute weight using equation 22.
Step 4. c1 and c2 are selected on the basis validation.
Step 5. Determine Lagrangian parameters α and β using eq. 45 and 47 and then calculate z_1 and z_2 using eq. 44 and 46.
Step 6. Determine two non-parallel kernel generated surfaces by using equation 41.
Step 7. Classify new patterns by using equation 48.

4. Results and Experiments

The performance of different classifiers is evaluated using confusion matrix. Confusion Matrix stores the details of actual and predicted class (estimated by the classifier) in tabular form as shown in table 1.

Table 1. Confusion Matrix

Actual Class ↓	Predicted Class →	
	Yes	No
Yes	True Positive (TP)	False Negative (FN)
No	False Positive (FP)	True Negative (TN)

This paper evaluates the performance of proposed methodology for class imbalance using accuracy, specificity, sensitivity and geometric mean. Accuracy of a classifier is estimated by the correct prediction made by the classifier in proportion to total number of prediction.

$$\text{Accuracy= (TP+TN)/(TP+FP+FN+TN)}$$

Sensitivity of a classifier is evaluated by the percentage of positive values that are recognized accurately. It is also known as true positive rate and formulated as:

$$\text{Sensitivity= TP/ (TP+FN)}$$

Specificity of a classifier is estimated by the percentage of negative values that are recognized correctly by the classifier. It is also known as true negative rate and formulated as:

$$\text{Specificity= TN/ (TN+FP)}$$

In this research paper, we also used other parameters for performance measure that is Geometric Mean. Geometric Mean is calculated by the sensitivity and specificity and formulated as:

$$\text{Geometric Mean=}\sqrt{(\text{Sensitivity} \times \text{Specificty})}$$

Sensitivity and specificity varies for each classifier. If we evaluate the performance of a classifier using these two parameters, it does not make an even comparison. So, to make effective comparison we calculate the Geometric Mean for each classifier and then evaluate their performance [28]. This research work performed experiments on five benchmark UCI datasets using 10-fold cross validation. The detailed descriptions of dataset are given in following Table 2:

Table 2. Details of Dataset used for Experiment

Dataset	Negative Sample	Positive Sample	No. of Features
Pima Indian Diabetes	500	268	8
Hepatitis	32	123	19
Sonar	97	111	60
Ionosphere	126	225	34
Heart-Statlog	150	120	13

All the above mentioned datasets are obtained from UCI Machine Learning Repository. Pima diabetes dataset contains 768 rows and 8 attributes among which 500 data samples are positive while remaining 268 data samples belong to negative class. PIMA Indian diabetes dataset is very popular dataset and suitable for testing of a prediction model. This dataset contains the information such as Glucose Level, Age, Body Mass Index, Blood Pressure etc. of pregnant women. This dataset is used to predict the gestational diabetes in pregnant woman. The size of Hepatitis dataset is 155 with 19 features among which 32 are negative data samples and 123 are positive data samples. Sonar dataset having 208 data samples with 60 features. 97 samples are collected from rocks while 111 samples are obtained from mines. This dataset stores different signal patterns collected from mines and rocks using different aspect angles. Ionosphere dataset contains 351 data samples with 34 features. This is the radar data which was obtained using a system in Goose Bay, Labrador. There are two categories of radar signals –one is good radar signals indicating some type of ionosphere structure while another one is bad radar signals that do not represent any type of ionosphere structure. There are 270 data samples in Heart Statlog dataset which contains the information regarding presence and absence of heart disease. It is clear that above mentioned dataset are imbalanced in nature i.e., the number of data samples belonging to different classes are different which influences the experimental results. In order to handle this situation, this research work performed experiments on above mentioned five benchmark datasets [30]. Proposed method is implemented using matlab R2012a. The performance of the proposed approach is tested using 10-fold cross validation and is also compared with traditional SVM. Results of experiment indicate that the proposed method obtains better classification accuracy as compared to other existing methods as well as it also handles the situation of imbalance dataset effectively. Table 3 and 4 indicate the performance comparison of different classifiers using Gaussian and Polynomial Kernel function. Accuracy, sensitivity, specificity and geometric mean parameters are used to evaluate the performance of WLSTSVM with other three existing methods such as SVM, TSVM and LSTSVM. The result of proposed work is also compared with under-sampling method and analysis regarding the performance of classifier with imbalanced data has been compared.

Table 3. Performance Comparison of Different Classifiers using Gaussian Kernel Function

Dataset	Performance Measurement Parameters	SVM	TSVM	Least Square TSVM	Weighted LS-TSVM	LS-TSVM after balancing data using Under Sampling
Pima Indian Diabetes	Accuracy	0.7589	0.7428	**0.7761**	0.7567	0.7431
	Sensitivity	0.5176	1.0000	0.8500	0.8709	0.8541
	Specificity	0.8875	0.2523	0.7103	0.6957	0.5432
	Geometric Mean	0.6778	0.5022	0.7770	**0.7783**	0.6811
Hepatitis	Accuracy	0.8213	0.8167	0.8428	**0.8667**	0.8333
	Sensitivity	0.8505	0.8495	0.8017	0.9167	1.0000
	Specificity	0.6729	0.6703	0.8723	0.6667	0.6667
	Geometric Mean	0.7565	0.7544	0.8362	**0.8913**	0.8165
Sonar	Accuracy	0.7626	0.7979	0.8304	**0.8571**	0.8421
	Sensitivity	0.6176	0.7626	0.8571	0.9091	0.7000
	Specificity	0.8875	0.8192	0.7879	0.8000	1.0000
	Geometric Mean	0.7403	0.7903	0.8217	**0.8528**	0.8367
Ionosphere	Accuracy	0.8604	0.8803	**0.8993**	0.8824	0.8400
	Sensitivity	0.8823	0.7169	0.7495	0.7267	0.8333
	Specificity	0.7934	0.9827	0.9579	1.0000	0.8462
	Geometric Mean	0.8366	0.8393	0.8473	**0.8524**	0.8397
Heart-Statlog	Accuracy	0.8407	0.8444	**0.8543**	0.8462	0.8479
	Sensitivity	0.8107	0.8681	0.8158	0.7673	0.7517
	Specificity	0.8685	0.8134	0.8936	0.9333	0.8693
	Geometric Mean	0.8391	0.8403	**0.8538**	0.8462	0.8083

Table 4. Performance Comparison of Different Classifiers using Polynomial Kernel Function

Dataset	Performance Measurement Parameters	SVM	TSVM	Least Square TSVM	Weighted LS-TSVM	LS_TSVM after balancing data using Under Sampling
Pima Indian Diabetes	Accuracy	0.7235	0.7559	**0.7800**	0.7693	0.7169
	Sensitivity	0.6471	0.7794	0.6333	0.7815	0.6031
	Specificity	0.8692	0.7196	1.0000	0.7174	0.8137
	Geometric Mean	0.7499	0.7489	**0.7958**	0.7487	0.7005
Hepatitis	Accuracy	0.8000	0.8079	0.8405	**0.8561**	0.8137
	Sensitivity	0.8505	0.7097	0.8065	0.8687	0.8419
	Specificity	0.6329	0.8567	0.8785	0.8234	0.6713
	Geometric Mean	0.7367	0.7797	0.8417	**0.8457**	0.7518
Sonar	Accuracy	0.7938	0.7963	0.8148	**0.8519**	0.8000
	Sensitivity	0.7603	0.7619	0.9048	1.0000	0.8000
	Specificity	0.8192	0.8182	0.7576	0.7000	0.8000
	Geometric Mean	0.7892	0.7895	0.8279	**0.8367**	0.8000
Ionosphere	Accuracy	0.8944	0.9061	0.9353	**0.9417**	0.8518
	Sensitivity	0.7182	0.7563	0.8571	0.8619	0.8137
	Specificity	1.0000	1.0000	0.9842	1.0000	0.8879
	Geometric Mean	0.8474	0.8696	0.9184	**0.9283**	0.8499
Heart-statlog	Accuracy	0.8214	0.8235	0.8313	**0.8647**	0.8333
	Sensitivity	0.9474	0.9079	0.9079	0.8553	0.9167
	Specificity	0.7021	0.7553	0.7766	0.8723	0.7500
	Geometric Mean	0.8156	0.8280	0.8396	**0.8637**	0.8292

From Table 3 and Table 4, it is clear that the proposed approach works well for imbalanced data. For Gaussian kernel function, the accuracy of WLSTSVM for Hepatitis and Sonar dataset is 86.67% and 85.71% which is better than other existing methods. For other datasets, the accuracy of proposed approach is also nearly comparable. While the Geometric Mean of the proposed approach is almost better for all datasets. The accuracy and geometric mean of WLSTSVM using Polynomial Kernel function are also better for Hepatitis, Sonar, Ionosphere and Heart-statlog dataset as compared to other existing methods. While for the Pima Indian Diabetes dataset, it is better than SVM, TSVM and under sampling approach.

5. Conclusion

Real world data are imbalanced in nature due to which most of the classification techniques do not work well. This research work proposes a novel Weighted Least Square Twin Support Vector Machine for imbalanced data. For exhaustive verification of the obtained outcome, proposed approach has been tested for multiple imbalanced dataset using Polynomial and Gaussian kernel function. The experiment shows that the proposed techniques perform well for imbalanced datasets and also its accuracy and geometric mean is better as compared to other existing methods. The result of the proposed work is also compared with that of under-sampling method and it is found to be better. So, this method could suitably be utilized as a predictive model for the classification of real world data which are imbalanced in nature.

References

[1] J. Laurikkala, “Instance-based data reduction for improved identification of difficult small classes”, Intell Data Anal, vol. 6, no.4, **(2002)**, pp. 311–322.

[2] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study”, Intelligent data analysis, vol. 6, no. 5, **(2002)**, pp. 429-449.

[3] M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets: one-sided selection”, Proceedings of the fourteenth international conference on machine learning, **(1997)**, pp. 179-186,

[4] C. Ling and C. Li, “Data mining for direct marketing—specific problems and solutions”, Proceedings of fourth international conference on knowledge discovery and data mining, **(1998)**, pp. 73-79.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique”, arXiv preprint arXiv:1106.1813, **(2011)**.

[6] P. Domingos, “MetaCost: a general method for making classifiers cost-sensitive”, Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, CA., **(1999)**, pp 155-164.

[7] C. Elkan, “The foundations of cost-senstive learning”, Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, WA, , pp 973-978, **(2001)**.

[8] K. M. Ting, “An instance-weighting method to induce cost-sensitive trees”, Knowledge and Data Engineering, IEEE Transactions, vol. 14, no. 3, **(2002)**, pp. 659-665.

[9] B. Zadrozny, J. Langford and N. Abe, “Cost-sensitive learning by cost-proportionate example weighting”, Proceedings of the 3rd IEEE international conference on data mining, Melbourne, FL, **(2003)**, pp. 435-442.

[10] Z.-H. Zhou and X.-Y. Liu, “Training cost-sensitive neural networks with methods addressing the class imbalance problem”, IEEE Trans Knowl Data Eng., vol. 18, no. 1, **(2006)**, pp. 63–77.

[11] R. Akbani, S. Kwek and N. Japkowicz, “Applying support vector machines to imbalanced datasets”, Machine Learning: ECML, 2004, Springer Berlin Heidelberg., **(2004)**, pp. 39-50.

[12] G. E. Batista, R. C. Prati and M. C. Monard, “A study of the behaviour of several methods for balancing machine learning training data”, ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, **(2004)**, pp. 20-29.

[13] B. X. Wang, “Boosting support vector machine”, Master Thesis, **(2005)**.

[14] B. X. Wang and N. Japkowicz, “Boosting Support Vector Machines for imbalanced data sets”, Knowledge and Information Systems, vol. 25, no. 1, **(2010)**, pp. 1-20.

[15] C. Chen, A. Liaw and L. Breiman, “Using random forest to learn imbalanced data”, University of California, Berkeley, **(2004)**.

[16] X. Yang, Q. Song and Y. Wang, “A weighted support vector machine for data classification”, International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, no. 05, **(2007)**, pp. 961-976.

[17] J. A. Suykens, J. De Brabanter, L. Lukas and J. Vandewalle, “Weighted least squares support vector machines: robustness and sparse approximation”, Neurocomputing, vol. 48, no. 1, **(2002)**, pp. 85-105.

[18] G. M. Fung and O. L. Mangasarian, “Multicategory proximal support vector machine classifiers”, Machine Learning, vol. 59, no. 1-2, **(2005)**, pp. 77-97.

[19] J. Chen and G. Ji, “Weighted least squares twin support vector machines for pattern classification”, Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on (Vol. 2, pp. 242-246). IEEE., **(2010)** February.

[20] C. Cortes and V. Vapnik, “Support Vector Network”, Mach Learn 20:273-297, **(1995)**.

[21] V. Vapnik, “The nature of statistical Learning”, @nd edn. Springer, New York, **(1998)**.

[22] N. Chistianini and J. Shawe-Taylor, “An Introduction to Support Vector Machines, and other kernel-based learning methods”, Cambridge University Press, **(2000)**.

[23] N. Cristianini and J. Shawe-Taylor, “An Introduction to Support Vector Machines”, Cambridge University Press**, (2000)**.

[24] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes”, Journal of Artificial Intelligence Research, vol. 2, **(1995)**, pp. 263–286.

[25] B. Sch¨olkopf, C. Burges and A. Smola, (Eds.), “Advances in Kernel Methods—Support Vector Learning”, MIT Press., **(1998)**.

[26] R. Jayadeva, R. Khemchandani and S. Chandra, “Twin Support vector Machine for pattern classification”, IEEE Trans Pattern Anal Mach Intell, vol. 29, no. 5, **(2007)**, pp. 905-910.

[27] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, “Improvements on Twin Support Vector Machines”, IEEE transactions on Neural Network, vol. 22, no. 6, **(2011)** June.

[28] T. Divya and A. Sonali, “Predictive Model for diabetic patients using Hybrid Twin Support Vector Machine”, 5th International Conferences on advances in communication Network and Computing (CNC-2014)., In Press.

[29] M. Arun Kumar and M. Gopal, “Least squares twin support vector machines for pattern classification”, Expert Systems with Applications, vol. 36, **(2009)**, pp. 7535–7543.

[30] Datatset. http://archive.ics.uci.edu/ml/datasets/.Accessed: 20th October 2013.

Authors

Divya Tomar, she is a research scholar in Information Technology Division of Indian Institute of Information Technology (IIIT), Allahabad, India under the supervision of Dr. Sonali Agarwal. Her primary research interests are Data Mining, Data Warehousing especially with the application in the area of Medical Healthcare.

Dr. Sonali Agarwal, is working as an Assistant Professor in the Information Technology Division of Indian Institute of Information Technology (IIIT), Allahabad, India. Her primary research interests are in the areas of Data Mining, Data Warehousing, E Governance and Software Engineering. Her current focus in the last few years is on the research issues in Data Mining application especially in E Governance and Healthcare.