Expert Systems with Applications 38 (2011) 8580–8585

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function

Jae Pil Hwang [1], Seongkeun Park [1], Euntai Kim *

School of Electrical and Electronic Engineering, Yonsei University, Sinchon-dong, Seodaemun-gu, Seoul 120-749, Republic of Korea

ARTICLE INFO

Keywords:
Imbalance dataset
Support vector machine
Lagrangian support vector machine

ABSTRACT

In this paper, a new weighted approach on Lagrangian support vector machine for imbalanced data classification problem is proposed. The weight parameters are embedded in the Lagrangian SVM formulation. The training method for weighted Lagrangian SVM is presented and its convergence is proven. The weighted Lagrangian SVM classifier is tested and compared with some other SVMs using synthetic and real data to show its effectiveness and feasibility.

1. Introduction

Since the first introduction of the support vector machine (SVM) Cortes & Vapnik, 1995, it has showed its usefulness in various applications such as classification (Chen & Wang, 2003), detection (Waring & Liu, 2005), pattern recognition (Pontil & Verri, 1998), etc. To improve the SVM, many researchers have worked on variations of SVM such as Fuzzy SVM (Lin & Wang, 2002; Lin & Wang, 2004), least square SVM (LS-SVM) Suykens & Vandewalle, 1999, total margin SVM (Fung & Mangasarian, 2001; Mangasarian & Musicant, 2001), Lagrangian SVM (LSVM) Mangasarian & Musicant, 2001 and proximal SVM (Fung & Mangasarian, 2001). These SVMs show their best performance when it's used for two-class problems with balanced training samples that have similar amount of training data in the positive training samples and negative training samples. In the real world, these training samples are not always balanced. For example, there exist more flowers that are not lily than flowers that are lily. SVM is known to have weakness in imbalanced training set. It shows good classification rate for the majority class. But the classification performance for the minority class deteriorates. 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 (Akbani, Kwek, & Japkowicz, 2004; Chawla, Bowyer, Hall, et al., 2002; Lessmann, 2004; Li, Qiao, & Liu, 2008; Liu, An, & Huang, 2006; Liu, Chen, & Lu, 2006; Tang, Zhang, Chawla, et al., 2009; Vilarino, Spyridonos, & Vitria, 2005; Wang & Zheng, 2008; Zou, Huang, & Wang, 2008).

In this paper, we propose a weighting method on LSVM to deal with the imbalanced data classification problem. LSVM has some better property compared to the standard SVM. First it can be learned iteratively which makes the training faster than using quadratic programming for training. Second LSVM determines the bias value in an analytic way. Although it has these benefits over SVM, it has not overcome the weakness in imbalanced problem. There has not been much work on improving LSVM for imbalance classification problem. In this paper, we added weight parameter to LSVM formulation so it can improve its performance for minority class with minimum influence on classification performance of majority class. We proved that with the weight, it can still be trained using the same method of LSVM.

This paper is organized as follows. In Section 2, we will briefly cover SVM and other methods applied to SVM for imbalanced classification problem. In Section 3, we derive weighted LSVM formulation and we present the training method and prove its convergence. In Section 4, we show the test result using synthetic dataset and real data from UCI repository (Frank & Asuncion, 2010). Finally, we deliver the conclusion in Section 5.

2. Preliminaries

2.1. Soft support vector machine

SVM tries to maximize the margin between the two classes so it can minimize the structural risk instead of empirical risk. Suppose a training set which consists of M data and has N features. The individual training datum is defined as $\mathbf{x}^i = [x_1^i, \ldots, x_N^i]^T$ and its class y^i. We define two matrices $\mathbf{Y} = diag(y^1, \ldots, y^M)$ and $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^M]^T$.

Assume a linear classifier with weight ω and bias b. We define two vectors, $\mathbf{e}$ and 0, which are filled 1 s and 0 s. The slack variable is defined as $\xi = [\xi^1, \ldots, \xi^M]^T$. The soft margin SVM is defined as

* Corresponding author. Tel.: +82 2 2123 2863.
E-mail addresses: purnnara@yonsei.ac.kr (J.P. Hwang), keiny@yonsei.ac.kr (S. Park), etkim@yonsei.ac.kr (E. Kim).
[1] Tel.: +82 2 21237729.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.01.061

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + C\mathbf{e}^T\xi, \\ &\text{s.t.} \quad \begin{cases} \mathbf{Y}(\mathbf{X}\boldsymbol{\omega} + \mathbf{e}b) - e + \xi \geqslant \mathbf{0}, \\ \xi \geqslant 0 \end{cases} \end{aligned} \tag{1}$$

and the solution can be obtained by using its dual formulation

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}, \\ &\text{s.t.} \quad \begin{cases} \mathbf{0} \leqslant \boldsymbol{\alpha} \leqslant C\mathbf{e}, \\ \mathbf{e}^T\mathbf{Y}\boldsymbol{\alpha} = 0 \end{cases} . \end{aligned} \tag{2}$$

To find the solution, we have to solve a quadratic constraint optimization problem.

2.2. SVM for imbalanced dataset

When the training data set is imbalanced, the difference between the classification performance of the majority set and minority set becomes larger. To overcome this problem, two methods have been proposed: one is based on sampling method and the other one is based on sample weighting method. First and foremost, sampling method can be divided into two classes: undersampling method and oversampling method. In the undersampling method, the training dataset is balanced by sampling the majority training set to make a smaller subset and make the size of the class even. Li et al. proposed a hybrid re-sampling method (Li et al., 2008) and Zou et al. proposed an undersampling method using genetic algorithm (Zou et al., 2008). Tang et al. reported a repetitive undersampling algorithm called GSVM-RU (granular SVMs-repetitive undersampling) (Tang et al., 2009). The undersampling method not only balances the positive and negative sample set but also lessens the training load of the SVM. But it is reported by Akbani that the undersampling might delete the crucial data and some valuable information might be lost (Akbani et al., 2004).

In the oversampling method, data from the minority class are copied multiple times or slightly changed such that the two classes are balanced. For example, syntheticminority over-sampling technique (SMOTE) was proposed in Chawla et al. (2002) in which some synthetic data are randomly generated in the interval between two minority samples. Some variants were also proposed such as (Akbani et al., 2004) in which the SMOTE and weighting approach were combined. The oversampling methods have proven their effectiveness in many research works (Akbani et al., 2004; Wang, 2008; Yang, Qiao, & Peng, 2007). But the oversampling methods also suffer from the increased computational cost due to the increase in the training data points.

The second approach to the imbalanced data classification problem is to apply the weights to the training data points (Huang & Du, 2005). The weighting method was combined with the SVM and many researchers have proposed a variety of weighted approaches. Yang et al. proposed weighted support vector machine for classification (Yang, Song, & Wang, 2007). Suykens proposed weighted LS-SVM (WLS-SVM) (Suykens, De Brabanter, Lukas, et al., 2002). Fung and Mangasarian applied weighted method to balance the training sample for proximal SVM for multiclass problem (Fung & Mangasarian, 2005). First we define the weight for each individual data as $w_i > 0$ and a weight matrix $\mathbf{W} = diag[w_1,\ldots,w_M]$. The primal problem of weighted SVM (WSVM) (Huang & Du., 2005) is defined as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + C\mathbf{e}^T\mathbf{W}\xi, \\ &\text{s.t.} \quad \begin{cases} \mathbf{Y}(\mathbf{X}\boldsymbol{\omega} + b\mathbf{e}) - \mathbf{e} + \xi \geqslant 0, \\ \xi \geqslant 0 \end{cases} . \end{aligned} \tag{3}$$

By giving the appropriate weight, this outperforms the standard SVM with imbalanced training set. Also there is weighted LS-SVM (Suykens et al., 2002), which the primal problem is defined as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{\mathbf{C}}{2}\xi^T\mathbf{W}\xi, \\ &\text{s.t.} \quad \mathbf{Y}(\mathbf{X}\boldsymbol{\omega} + b\mathbf{e}) - \mathbf{e} + \boldsymbol{\xi} = 0. \end{aligned} \tag{4}$$

3. A weighted approach to imbalanced dataset classification by Lagrangian support vector machine

3.1. Deciding the weight for imbalanced problem

Setting the appropriate weight is a critical issue in weighted approach for imbalanced problem. The weight has to fulfill two conditions. First the data in the majority class have to receive lower weight than those in the minority class receives. Second, the weight should satisfy $w_i \epsilon (0,1]$ so that the WLSVM can be trained with convergence. The justification for the boundary of the weight will be presented in Section 3.3. To deal with imbalanced dataset, we simply set the weight according to the size of positive and negative dataset. When the size of positive set is N_{pos} and that of negative set is N_{neg}, the weights are defined as

$$w_i = \begin{cases} 1/N_{pos} & \text{if } y_i = 1, \\ 1/N_{neg} & \text{otherwise}. \end{cases} \tag{5}$$

This method is most commonly used when deciding the weight of imbalanced training dataset. In our algorithm, the convergence speed is faster when the weight is large. To maintain the weight ratio and make the convergence speed faster, we use the following weighting formula (6) instead of (5)

$$w_i = \begin{cases} 1 & \text{if } y_i = 1 \text{ and } N_{pos} \geq N_{neg}, \\ N_{neg}/N_{pos} & \text{if } y_i = 1 \text{ and } N_{pos} < N_{neg}, \\ N_{pos}/N_{neg} & \text{if } y_i = -1 \text{ and } N_{pos} \geq N_{neg}, \\ 1 & \text{if } y_i = -1 \text{ and } N_{pos} < N_{neg}. \end{cases} \tag{6}$$

3.2. Problem formulation

The LSVM applies quadratic cost function on the slack variable and the bias term is also included in the object function. LSVM is formulated as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{1}{2}b^2 + \frac{C}{2}\xi^T\xi, \\ &\text{s.t.} \quad \begin{cases} \mathbf{Y}(\mathbf{X}\boldsymbol{\omega} + b\mathbf{e}) - e + \xi \geqslant \mathbf{0}, \\ \xi \geqslant 0. \end{cases} \end{aligned} \tag{7}$$

To deal with the imbalanced dataset, a weight w^i is assigned to each data point to (7) and the weighted LSVM is formulated as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{1}{2}b^2 + \frac{C}{2}\xi^T\mathbf{W}\xi, \\ &\text{s.t.} \quad \begin{cases} \mathbf{Y}(\mathbf{X}\boldsymbol{\omega} + b\mathbf{e}) - \mathbf{e} + \xi \geqslant \mathbf{0}, \\ \xi \geqslant 0. \end{cases} \end{aligned} \tag{8}$$

First, the constraint condition $\xi \geqslant 0$ is removed from the formulation as in Mangasarian and Musicant (2001). Its Lagrangian is formulated as

$$L = \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{1}{2}b^2 + \frac{C}{2}\xi^T\mathbf{W}\xi - \boldsymbol{\alpha}^T(\mathbf{Y}(\mathbf{X}\boldsymbol{\omega} + b\mathbf{e}) - \mathbf{e} + \xi) \tag{9}$$

using the multiplier $\boldsymbol{\alpha} \geqslant 0$. The optimal point of (9) is located at the saddle point where in the aspect of ξ, $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$. Zeroing the derivative L of with respect to the primal variables yields

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\omega}} &= \boldsymbol{\omega} - \mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha}, \\ \frac{\partial L}{\partial b} &= b - \mathbf{e}^T\mathbf{Y}\boldsymbol{\alpha}, \\ \frac{\partial L}{\partial \boldsymbol{\xi}} &= \boldsymbol{\alpha} - C\mathbf{W}\xi. \end{aligned} \tag{10}$$

The primal variables $\boldsymbol{\omega}$, b and ξ are obtained by

$$\begin{aligned}\boldsymbol{\omega} &= \mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha},\\ b &= \mathbf{e}^T\mathbf{Y}\boldsymbol{\alpha}\\ \xi &= C\mathbf{W}^{-1}\boldsymbol{\alpha}.\end{aligned} \tag{11}$$

As can be seen in (11), ξ is automatically positive and the removal of the constraint $\xi \geqslant 0$ is justified. By substituting the primal variables in (9) with the right term of (11), we obtain the dual formulation of (8) as

$$\begin{aligned}&\text{minimize} \quad \tfrac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{Y}(\mathbf{X}\mathbf{X}^T + \mathbf{e}\mathbf{e}^T)\mathbf{Y} + \tfrac{1}{C}\mathbf{W}^{-1}\right)\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha},\\ &\text{s.t.} \quad\quad \boldsymbol{\alpha} \geqslant 0.\end{aligned} \tag{12}$$

3.3. Training the weighted Lagrangian support vector machine

In this subsection, the training rules for the weighted LSVM will be derived. The derivation is very similar to that of the plain LSVM (Mangasarian & Musicant, 2001) but it is modified such that the sample weights are accommodated. For the sake of notational simplicity, we define new matrices $\mathbf{Q} = \frac{1}{C}\mathbf{W}^{-1} + \mathbf{H}\mathbf{H}^T$ and $\mathbf{H} = \mathbf{Y}[\mathbf{X} - \boldsymbol{e}]$. As in Mangasarian and Musicant (2001), the iterative equation

$$\boldsymbol{\alpha}^{i+1} = \mathbf{Q}^{-1}(\mathbf{e} + ((\mathbf{Q}\boldsymbol{\alpha}^i - \mathbf{e}) - \gamma\boldsymbol{\alpha}^i)_+), \quad i = 0, 1, \ldots. \tag{13}$$

is used to train the weighted LSVM where $\boldsymbol{\alpha}^i$ is the calculated value of the Lagrangian multiplier after ith iteration. Since each datum has its own weight, an appropriate value of γ should be determined such that the training convergence of (13) is guaranteed. The next theorem proves the theoretical bound of γ which guarantees the convergence of the algorithm.

Theorem 1. *If γ satisfies the condition*

$$0 < \gamma < \frac{2}{C} \tag{14}$$

and the boundary of the weight is defined as $w^i \epsilon (0,1]$, then Eq. (13) converges to the optimal point $\tilde{\mathbf{u}}$.

Proof. The proof is similar to that of Mangasarian and Musicant (2001). If

$$\|\mathbf{Q}\boldsymbol{\alpha}^{i+1} - \mathbf{Q}\tilde{\boldsymbol{\alpha}}\| < \|\mathbf{Q}\boldsymbol{\alpha}^{i} - \mathbf{Q}\tilde{\boldsymbol{\alpha}}\| \tag{15}$$

is satisfied for $\forall \boldsymbol{\alpha}^{i+1}, \boldsymbol{\alpha}^i \neq \tilde{\boldsymbol{\alpha}}$, then $\boldsymbol{\alpha}^i$ converges to $\tilde{\boldsymbol{\alpha}}$. Substituting (13) to the left side of (15) yields

$$\begin{aligned}\|\mathbf{Q}\boldsymbol{\alpha}^{i+1} - \{\mathbf{Q}\tilde{\boldsymbol{\alpha}}\| &= \|(\mathbf{Q}\boldsymbol{\alpha}^i - \mathbf{e} - \gamma\boldsymbol{\alpha}^i)_+ - (\mathbf{Q}\tilde{\boldsymbol{\alpha}})_+\|\\ &\leq \|(\mathbf{Q} - \gamma\mathbf{I})(\boldsymbol{\alpha}^i - \tilde{\boldsymbol{\alpha}})\|\\ &\leq \|\mathbf{I} - \gamma\mathbf{Q}^{-1})\|\|(\mathbf{Q}\boldsymbol{\alpha}^i - \mathbf{Q}\tilde{\boldsymbol{\alpha}})\|\end{aligned} \tag{16}$$

If γ is selected such that

$$\|\mathbf{I} - \gamma\mathbf{Q}^{-1}\| < 1, \tag{17}$$

(15) is obtained and the algorithm converges. Using the eigenvalue decomposition, $\mathbf{Q}$ is represented as $\mathbf{E}^T\boldsymbol{\lambda}\mathbf{E}$ where $\boldsymbol{\lambda} = diag(\lambda_1, \ldots, \lambda_N)$ and $\mathbf{E}$ is a unitary matrix. Then,

$$\begin{aligned}\|\mathbf{I} - \gamma\mathbf{Q}^{-1}\| &= \left\|\mathbf{I} - \gamma(\mathbf{E}^T\boldsymbol{\lambda}\mathbf{E})^{-1}\right\| = \|\mathbf{E}^T\mathbf{E} - \gamma\mathbf{E}^T\lambda^{-1}\mathbf{E}\|\\ &= \|\mathbf{E}\|\|\mathbf{I} - \gamma\lambda^{-1}\|\|\mathbf{E}\| = \|\mathbf{I} - \gamma\lambda^{-1}\|.\end{aligned} \tag{18}$$

To satisfy Eq. (17), the absolute value of the diagonal component of $\mathbf{I} - \gamma\lambda^{-1}$ is less than 1. This can be rewritten this statement into

$$-1 < 1 - \frac{\gamma}{\lambda_{min}(\mathbf{Q})} < 1 - \frac{\gamma}{\lambda_{min}(\mathbf{Q})} < 1. \tag{19}$$

By adding and subtracting 1 to both sides and multiplying -1, (19) becomes

$$0 < \frac{\gamma}{\lambda_{min}(\mathbf{Q})} < \frac{\gamma}{\lambda_{min}(\mathbf{Q})} < 2. \tag{20}$$

From (20) we can obtain the following condition

$$0 < \gamma < 2\lambda_{min}(\mathbf{Q}) = 2\lambda_{min}\left(\frac{1}{C}W^{-1} + \mathbf{H}\mathbf{H}^T\right). \tag{21}$$

Here, from the definition of $\mathbf{W}^{-1}$ that all its eigenvalues are bigger than 1 and the positive semidefiniteness of $\mathbf{H}\mathbf{H}^T$, it is clear that

$$\frac{1}{C}\mathbf{I} \leq \frac{1}{C}\mathbf{W}^{-1}\frac{1}{C}\mathbf{W}^{-1} + \mathbf{H}\mathbf{H}^T \tag{22}$$

and

$$\frac{1}{C} \leq \lambda_{min}\left(\frac{1}{C}\mathbf{W}^{-1} + \mathbf{H}\mathbf{H}^T\right). \tag{23}$$

From (21)–(23), if γ is selected by

$$0 < \gamma < \frac{2}{C}, \tag{24}$$

then (15) and (17) are satisfied and the iterative method converges QED.

3.4. Weighted Lagrangian support vector machine with nonlinear kernel

In this subsection, the kernel version of weighted LSVM is introduced to deal with nonlinear case. Let us define the kernel function $K(\mathbf{A}, \mathbf{B}) = \varphi(\mathbf{A})\varphi(\mathbf{B})^T$. This kernel function maps an input $\mathbf{A} \in \Re^{n\times m}$ and $\mathbf{B} \in \Re^{l\times m}$ to a $\Re^{n\times l}$ space. For example, in the case of the linear kernel, $\varphi(\mathbf{A}) = \mathbf{A}$ and $K(\mathbf{A}, \mathbf{B}) = \mathbf{A}\mathbf{B}^T$. For the Gaussian kernel, (i,j)th component of $K(\mathbf{A}, \mathbf{B})$ is $\varepsilon^{-(\mathbf{A}_i - \mathbf{B}_j)/\sigma^2}$ where ε is the base of natural logarithm and $\mathbf{A}_i$ and $\mathbf{B}_j$ are the ith and jth row vector of each matrix respectively. We use the formulation

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{1}{2}b^2 + \frac{C}{2}\xi^T\mathbf{W}\xi,\\ &\text{s.t.} \quad\quad \begin{cases}\mathbf{Y}(\varphi(\mathbf{X})\boldsymbol{\omega} + b\mathbf{e}) - \mathbf{e} + \xi \geq 0,\\ \xi \geq 0.\end{cases}\end{aligned} \tag{25}$$

By using Lagrangian method, the dual formulation of (25) becomes

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{Y}(K(\mathbf{X},\mathbf{X}) + \mathbf{e}\mathbf{e}^T)\mathbf{Y} + \frac{1}{C}\mathbf{W}^{-1}\right)\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha},\\ &\text{s.t.} \quad\quad \boldsymbol{\alpha} \geq 0.\end{aligned} \tag{26}$$

and the classifier becomes

$$y = \text{sgn}(K(\mathbf{x}^T, \mathbf{X})\mathbf{Y}\mathbf{a} + b), \tag{27}$$

where b is defined as in Eq. (11). The training method is identical to that in (13) except $\mathbf{Q} = \frac{1}{C}\mathbf{W}^{-1} + \mathbf{Y}(K(\mathbf{X},\mathbf{X}) + \mathbf{e}\mathbf{e}^T)\mathbf{Y}$.

4. Simulation

In this section we explain the metric for performance measurement. Then we present the simulation result using synthetic data and real data that is taken from UCI repository.

4.1. Performance metric for imbalance problem

The result of classification can be categorized into four cases which are stated in the Table 1, confusion matrix. These categories are called TP (true positive), FN (false negative), FP (false positive) and TN (true negative). We define the sizes of the samples that belong to each categories as N_{TP}, N_{FN}, N_{FP} and N_{TN}.

To measure the performance of the classifier, it is not fair just to compare the overall accuracy P_{acc} which is defined as

Table 1
Confusion matrix.

	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

$$P_{acc} = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{FP} + N_{FN} + N_{TN}}. \quad (28)$$

Because the size of the negative samples is much larger than that of the positive samples, the influence of the negative sample is much higher than that of positive samples. For this reason, accuracy is not a good metric for this case. To make an even comparison of performance, we use geometric mean (G-Mean) instead of accuracy (Bradley, 1997). G-Mean is defined by two parameter called sensitivity and specificity. Sensitivity shows the performance of the positive class and specificity shows the performance of the negative class. Sensitivity P_{sens} and specificity P_{spec} are defined as in Eqs. (29) and (30)

$$P_{sens} = N_{TP}/(N_{TP} + N_{FN}), \quad (29)$$

$$P_{spec} + N_{TN}/(N_{TN} + N_{FP}). \quad (30)$$

G-Mean $P_{G\text{-Mean}}$ is defined as

$$P_{G\text{-Mean}} = \sqrt{P_{sens} \times P_{spec}} \quad (31)$$

which gives a more fair comparison of positive class and negative class regardless of its size.

Table 2
Synthetic dataset.

Data set	Center	Variance
Positive set	(2, 0)	$\begin{pmatrix} 1.5^2 & 0 \\ 0 & 1.5^2 \end{pmatrix}$
Negative set	(−2, 0)	$\begin{pmatrix} 1.5^2 & 0 \\ 0 & 1.5^2 \end{pmatrix}$

4.2. Synthetic data

We conducted the test on synthetic dataset which has two features. We have generated 150 positive samples and combined with 160, 200, 300, 600, 1050 negative samples using the parameters in Table 2. For testing, we have generated 2000 positive test samples and negative samples along rational to size of the training set using the same parameter that is defined as in Table 2. The numbers in bold show the best accuracy and G-mean with the same sample size.

We have compared with 4 other different classifiers along with linear kernel. We used SVM, LS-SVM and LSVM which are not a weighted approach. The other classifier is WLS-SVM which is also a weighted approach. All these classifiers are programmed using MATLAB. The SVM was programmed using quadratic programming function. Unlike SVM, LSVM and our method, LS-SVM and WLSVM consider all the points in the training data. To find the optimal C, we tested these on various values from 100 down to 0.01.

In the aspect of accuracy, the SVM, LSVM and LS-SVM show better performance in most cases which uses weighted approach. On the other hand, when the data is imbalanced, G-mean of the WLSVM, WLS-SVM is better than approaches without weights. We can see the decline of the G-mean when the imbalance increases. On the other hand, the classifiers embedding the weight shows almost constant performance compared to other method. In most of the cases, WLSVM outperformed WLS-SVM. There were some cases that WLS-SVM outperformed WLSVM. We think that

Table 4
Dataset summery that is used for test.

Dataset	Size of positive sample	Size of negative sample	Ratio (majority/minority)	Feature size
Pima Indian diabetes	268	500	1.87	8
Landsat satellite image	626	5809	9.28	36

Table 3
Test results using synthetic dataset.

Classifier	Positive sample size	Negative sample size	C	Sensitivity	Specificity	Accuracy	G-Mean	Training Time	Number of SVs
SVM	150	160	0.1	0.93040	0.93690	**0.93375**	**0.93364**	29.39368	147
	150	200	0.1	0.92570	0.94080	**0.93433**	0.93322	28.76426	175
	150	300	0.1	0.92790	0.93880	0.93517	**0.93333**	88.22618	262
	150	600	0.03	0.94190	0.92160	0.92566	0.93169	350.36975	553
	150	1050	0.01	0.94910	0.91080	0.91559	0.92975	210.45702	1016
LS-SVM	150	160	3	0.93110	0.93580	0.93353	0.93345	0.03671	NA
	150	200	0.1	0.91610	0.94760	0.93410	0.93172	0.05471	NA
	150	300	1	0.91260	0.95100	**0.93820**	0.93160	0.10838	NA
	150	600	30	0.81010	0.98330	**0.94866**	0.89251	0.43046	NA
	150	1050	30	0.65470	0.99490	**0.95238**	0.80707	1.46172	NA
LSVM	150	160	0.1	0.93580	0.93030	0.93296	0.93305	0.11224	140
	150	200	3	0.94060	0.92220	0.93009	0.93135	0.17228	134
	150	300	30	0.94750	0.91360	0.92490	0.93040	0.37999	133
	150	600	10	0.97370	0.84660	0.87202	0.90793	1.75097	150
	150	1050	10	0.98370	0.79150	0.81553	0.88238	3.85461	243
Weighted LS-SVM	150	160	10	0.93300	0.93380	0.93341	0.93340	0.05973	NA
	150	200	0.1	0.93170	0.93430	0.93319	0.93300	0.08734	NA
	150	300	0.03	0.94430	0.91830	0.92697	0.93121	0.18762	NA
	150	600	10	0.93370	0.93290	0.93306	0.93330	0.77741	NA
	150	1050	3	0.93030	0.93640	0.93564	**0.93335**	2.73650	NA
Weighted LSVM	150	160	0.03	0.93350	0.93320	0.93335	0.93335	0.12069	185
	150	200	3	0.93090	0.93620	0.93393	**0.93355**	0.19999	136
	150	300	0.01	0.92440	0.94190	0.93607	0.93311	0.27536	367
	150	600	0.01	0.93360	0.93320	0.93328	**0.93340**	0.98479	598
	150	1050	0.01	0.93400	0.93240	0.93260	0.93320	3.17964	998

Table 5
Test results using real dataset from UCI repository.

Dataset	Classifier	Parameters σ, C	Sensitivity	Specificity	Accuracy	G-Mean	Training time	Number of SVs
Pima Indian diabetes	SVM	1024, 512	0.52872 (0.036438)	0.89100 (0.042544)	0.764628 (0.025901)	0.685623 (0.024181)	1.381591 (0.295129)	252.6 (15.18186)
	LS-SVM	256, 32	0.72192 (0.067457)	0.781173 (0.015793)	0.764301 (0.025607)	0.750342 (0.038922)	0.838208 (0.345853)	N/A
	LSVM	256, 8	0.716723 (0.053294)	0.782049 (0.01386)	0.763656 (0.020114)	0.685623 (0.024181)	3.673777 (0.182471)	440.1 (4.840799)
	WSVM	1024, 257	0.682945 (0.041204)	0.7735 (0.03317)	0.741852 (0.027001)	0.726406 (0.028032)	1.348873 (0.206039)	280.9 (14.402546)
	WLS-SVM	512, 128	0.81376 (0.030763)	0.743608 (0.010637)	0.754562 (0.007951)	0.777684 (0.01126)	0.685421 (0.011188)	N/A
	WLSVM	512, 32	0.816451 (0.033494)	0.737505 (0.01386)	0.749038 (0.011745)	0.775745 (0.014122)	4.055852 (0.101997)	432.0 (7.272475)
Landsat satellite image	SVM	32, 4	0.635532 (0.033885)	0.983302 (0.001912)	0.949458 (0.003772)	0.790265 (0.021014)	8.517899 (0.251889)	1233.9 (18.351203)
	LS-SVM	32, 8	0.809708 (0.019952)	0.960411 (0.00364)	0.949069 (0.003386)	0.881785 (0.011087)	78.776061 (4.840682)	N/A
	LSVM	32, 8	0.798449 (0.015556)	0.961844 (0.003037)	0.949107 (0.002821)	0.876309 (0.008763)	87.12359 (0..969109)	1670.6 (41.06688)
	WSVM	64, 16	0.772223 (0.052213)	0.946763 (0.004867)	0.92976 (0.002178)	0.85455 (0.026834)	19.610528 (1.492259)	686.8 (27.251096)
	WLS-SVM	32, 16	0.923927 (0.02318)	0.942583 (0.002153)	0.941687 (0.002281)	0.933142 (0.011792)	105.254154 (9.92494)	N/A
	WLSVM	32, 8	0.944761 (0.018659)	0.93642 (0.001858)	0.936713 (0.001868)	0.940539 (0.009271)	84.006022 (0.636945)	1638.2 (20.757863)

these results were originated by the distribution of the training set and the difference of the method, which involves all data and part of the data Table 3.

4.3. Test using UCI database

We have test our performance on 3 different datasets from UCI dataset (Frank & Asuncion, 2010). The dataset is summarized in Table 4.

The Pima dataset was collected by National Institute of Diabetes and Digestive and Kidney Diseases. This dataset collected the medical record of patients with the diabetes. This has 268 positive samples and 500 negative samples with 8 features. The Landsat Satellite image dataset consist of 6 classes originally. We select the class with least samples and take it as the positive samples. The rest of the samples are treated as negative samples. This dataset has 36 features which is 3 by 3 pixel image with 4 different channels.

Ten-fold verification is used for testing the classifiers. Four tenth of the data is used for training and the rest are used for testing. SVMLight (Joachims, 1999) is used for testing SVM and WSVM. LS-SVM, LSVM, WLS-SVM and WLSVM are programmed using MATLAB. The test results are summarized in Table 5. The value in each cell outside the parenthesis is the mean of the ten trials and the value inside the parenthesis is the standard deviation of them. Because LS-SVM and WLS-SVM utilize all the training data to make the classifier, the number of support vectors for LS-SVM and WLS-SVM are not stated in Table 5.

The methods that are not using weight parameter, such as SVM, LSVM and LS-SVM, shows better accuracy compared to the methods using weight. But these methods generate large difference between sensitivity and specificity, causing the degradation in G-mean. The methods embedding weight shows better G-mean then ones that are not. Among these three methods, WLS-SVM shows similar performance compared to WLSVM and WSVM showed the lowest G-mean. But the WLS-SVM uses all the training samples and needs more time for actual verification. WLSVM uses only part of the training samples and shows similar G-Mean compared to WLS-SVM. The training time of the classifiers programmed using MATLAB was higher than that of SVM and WSVM. These results are caused by the time delay for full matrix inversion.

5. Conclusion and further research

In this paper we proposed weighted approach of LSVM to deal with imbalanced problem. We have added weight on the LSVM formulation. We also showed that on certain condition, the WLSVM could be trained using the same method as standard LSVM. This method improved the performance on imbalanced problem for LSVM.

The training time was faster than SVM using quadratic programming. But it was slower than SVM using SMO algorithm. Our future work will concentrate on improving the training method so that it can work faster. We think there's much room for improvement.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0012631).

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Machine Learning: ECML*, 39–50.
Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159.
Chawla, N. V., Bowyer, K. W., Hall, L. O., et al. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.
Chen, Y. X., & Wang, J. Z. (2003). Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems, 11*(6), 716–728.
Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.
Frank, A. & Asuncion, A. (2010). *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>.
Fung, G., & Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2001)* (pp. 77–86).
Fung, G. M., & Mangasarian, O. L. (2005). Multicategory proximal support vector machine classifiers. *Machine Learning, 59*(1–2), 77–97.

Huang, Y. M., & Du, S. X. (2005). Weighted support vector machine for classification with uneven training class sizes. In *International conference on machine learning and cybernetics, ICMLC* (pp. 4365–4369).

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods - support vector learning*. MIT-Press.

Lessmann, S. (2004). Solving imbalanced classification problems with support vector machines. In *Proceedings of the international conference on artificial intelligence, IC-AI'04* (pp. 214–220).

Li, P., Qiao, P. L., & Liu, Y. C. (2008). A hybrid re-sampling method for SVM learning from imbalanced data sets. In *Fifth international conference on fuzzy systems and knowledge discovery, FSKD 2008* (pp. 65–69).

Lin, C. F., & Wang, S. D. (2002). Fuzzy support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 464–471.

Lin, C. F., & Wang, S. D. (2004). Training algorithms for fuzzy support vector machines with noisy data. *Pattern Recognition Letters, 25*(14), 1647–1656.

Liu, Y. H., Chen, Y. T., & Lu, S. S. (2006). Face detection using kernel PCA and imbalanced SVM. *Advances in Natural Computation, 4221*(Pt. 1), 351–360.

Liu, Y., An, A. J., & Huang, X. J. (2006). Boosting prediction accuracy on imbalanced datasets with SVM ensembles. In *Advances in knowledge discovery and data mining, proceedings*, (Vol. 3918, pp. 107–118).

Mangasarian, O. L., & Musicant, D. R. (2001). Lagrangian support vector machines. *Journal of Machine Learning Research, 1*(3), 161–177.

Pontil, M., & Verri, A. (1998). Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(6), 637–646.

Suykens, J. A. K., De Brabanter, J., Lukas, L., et al. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing, 48*, 85–105.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*(3), 293–300.

Tang, Y. C., Zhang, Y. Q., Chawla, N. V., et al. (2009). SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics, 39*(1), 281–288.

Vilarino, F., Spyridonos, P., Vitria, J. et al. (2005). Experiments with SVM and stratified sampling with an imbalanced problem: Detection of intestinal contractions. In *Pattern Recognition and Image Analysis, Pt. 2, Proceedings* (Vol. 3687, pp. 783–791).

Wang, H., & Zheng, H. (2008). An improved support vector machine for the classification of imbalanced biological datasets. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). In *Fourth international conference on intelligent computing, ICIC 2008* (pp. 63–70).

Wang, H. Y. (2008). Combination approach of SMOTE and biased-SVM for imbalanced datasets. In *2008 International joint conference on neural networks, IJCNN 2008* (pp. 228–231).

Waring, C. A., & Liu, X. W. (2005). Face detection using spectral histograms and SVMs. *IEEE Transactions on Systems Man and Cybernetics Part B – Cybernetics, 35*(3), 467–476.

Yang, Z. M., Qiao, L. Y., & Peng, X. Y. (2007). Research on data mining method for imbalanced dataset based on improved SMOTE. *Tien Tzu Hsueh Pao/Acta Electronica Sinica, 36*(Suppl. 2), 22–26.

Yang, X., Song, Q., & Wang, Y. (2007). A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence, 21*(5), 961–976.

Zou, S., Huang, Y., Wang, Y. et al. (2008). SVM learning from imbalanced data by GA sampling for protein domain prediction. In *Ninth international conference for young computer scientists, ICYCS 2008* (pp. 982–987).