Building a Biased Least Squares Support Vector Machine Classifier for Positive and Unlabeled Learning

Ting Ke, Lujia Song, Bing Yang, Xinbin Zhao, Ling Jing
Department of Applied Mathematics, College of Science, China Agricultural University, 100083, Beijing, P.R. China
Email: kk.ting@163.com, biggirlsong@126.com, 434915607@qq.com, zhaoxinbin1986811@163.com, jingling@cau.edu.cn

***Abstract*—Learning from positive and unlabeled examples (PU learning) is a special case of semi-supervised binary classification. The key feature of PU learning is that there is no labeled negative training data, which makes the traditional classification techniques inapplicable. Similar to the idea of Biased-SVM which is one of the most famous classifier, a biased least squares support vector machine classifier (Biased-LSSVM) is proposed for PU learning in this paper. More specifically, we take unlabeled examples as negative examples with noise and build a least squares support vector machine classifier using two penalty parameters C_p and C_n to weight misclassification errors of positive and negative examples respectively. As we pay more attention to classify as many as positive examples correctly in PU learning, the relationship of parameters C_p and C_n is $C_p \geq C_n$. Compared with Biased-SVM, the proposed classifier has three advantages. First, Biased-LSSVM can reflect the class labels of all examples more sufficiently and accurately than Biased-SVM. Second, Biased-LSSVM is more stable than Biased-SVM because the performance of Biased-LSSVM changes less than that of Biased-SVM over a wide ratio of positive examples in unlabeled examples. Finally, the time complexity of Biased-LSSVM is lower than that of Biased-SVM, where Biased-LSSVM only need to solve liner equations and Biased-SVM is a quadratic programming. The Experiments on two real applications, text classification and bioinformatics classification verify the above opinions and show that Biased-LSSVM is more effective than Biased-SVM and other popular methods, such as EB-SVM, ROC-SVM and S-EM.**

***Index Terms*—positive and unlabeled learning, least squares support vector machine, text classification, bioinformatics classification**

I. Introduction

Traditional machine learning techniques require a large number of labeled examples to learn an accurate classifier. This approach to building classifiers is called supervised learning. However, in many practical classification applications such as document retrieval and classification, negative examples are either hard to obtain or even not available at all, but plentiful unlabeled examples can acquire easily. In such cases, an algorithm for classification that only exploits positive and unlabeled examples is needed. We call this problem as partially supervised learning or PU learning (learn from positive and unlabeled examples) which is a special case of semi-supervised learning actually [1-3].

Denis originally proposes a framework for learning model from positive examples based on the probably approximately correct model (PAC) [4]. Learning from positive example was also studied theoretically in [5] within a Bayesian framework where the distribution of functions and examples were assumed known. It was shown in [6] that if the sample size was large enough, minimizing the number of unlabeled examples classified as positive while constraining the positive examples to be correctly classified would give a good classifier.

Recently, various approaches had been suggested in the literatures to solve PU learning. These algorithms include 1-DNF [7], ROC-SVM [8], PNLH [9], LCLC [10], SPUL [11], en-LCLC [12]. These methods were two-step strategy, which selected possible negative or positive examples from unlabeled examples, and then built classifiers using positive examples and negative examples. In addition, probability estimation approach, such as [13, 14], was proposed. In [14], it took each unlabeled example as both positive and negative example with weights pre-computed by an additional classifier for protein record identification. Luigi Cerulo et al made use of the approach in [14] to reconstruct gene regulatory networks without negative examples [15]. In fact, the most popular technique for PU learning was Biased-SVM [16]. It was built by giving appropriate weights to the positive examples and unlabeled examples respectively; here the unlabeled examples were regarded as negative examples with noise. Experimental results indicated that the performance was better than most of two-step strategies in text classification. In addition, WL [17] used Logistic Regression after weighting the negative class. And EB-SVM [18] was the improvement on Biased-SVM by giving an extra penalty parameter for some reliable positive examples. There were some other methods.

Corresponding author: Ling Jing
E-mail address: jingling@cau.edu.cn (L. Jing)
Tel: +86-10-62736511

doi:10.4304/jsw.9.6.1494-1502

Letouzey et al [19] presented an algorithm for learning using a modified C4.5 algorithm based on statistical query model. In [20], it needed information about the ratio of positive examples in unlabeled examples to solve the problem. LP-IC [21] devised an iterative classification scheme and introduced a class dispersion measure. A bagging SVM approach was first proposed for PU learning [22]. And an active learning algorithm was proposed in [23].

Because of the theoretical guarantee in [6], similar to the idea of Biased-SVM, we construct a biased least squares support vector machine classifier (Biased-LSSVM) by giving two different parameters to weight the errors of the positive examples and unlabeled examples respectively, here we also regard the unlabeled examples as negative examples with noise. Intuitively, we give a large parameter value for the errors of positive examples and a small parameter value for the errors of negative examples because negative examples also contain some positive examples. Experiments on two real applications, text classification and bioinformatics classification show that Biased-LSSVM is more effective than Biased-SVM, EB-SVM and other popular two-step methods, such as ROC-SVM and S-EM.

The rest of this paper is organized as follows. Some previous works are introduced in Section II. The proposed Biased-LSSVM is presented in Section III. Then in Section IV, experiments on two real-world applications are reported. Finally, we conclude this paper in Section V.

II. Related Works

A. Least Squares Support Vector Machine

The least squares support vector machine is a promising classification technique proposed firstly by Suykens et al [24]. Similar to support vector machine [22], least squares SVM also seeks an optimal separating hyper-plane that maximizes the margin between two classes. Consider a binary classifier, which uses a hyper-plane $g(x) =< w, \varphi(x) > +b = 0, w \in \mathbb{R}^n, b \in \mathbb{R}$ to separate two classes based on given training examples $\{x_i, y_i\}, i = 1, \cdots, l$, where x_i is a vector in the input space $\mathbb{R}^n$ and y_i denotes the class label taking a value of +1 or -1. $\varphi(\cdot)$ is a nonlinear function which is used to map the input space into a higher dimensional space. The least squares SVM solution is obtained through maximizing the margin between the separating hyper-plane and the data, where the margin is defined as $2/\|w\|$. Therefore, the maximum-margin classifier is obtained by solving

$$\min_{(w,b,\xi)} \quad \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2$$

$$s.t.\ \ y_i(< w, \varphi(x_i) > +b) = 1 - \xi_i \quad i = 1,2,...,l \qquad (1)$$

Introducing the Lagrange multipliers α_i from optimization problem (1), we obtain

$$L(w,b,\xi,\alpha) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i(y_i(\varphi(x_i)\cdot w + b) - 1 + \xi_i) \qquad (2)$$

Taking the partial derivatives of (2) with respect to w, b, ξ_i, α_i and equating them to zero, we obtain the optimal conditions as follows

$$\frac{\partial L}{\partial w} = 0 \to w = \sum_{i=1}^{l}\alpha_i y_i \varphi(x_i) \qquad (3)$$

$$\frac{\partial L}{\partial b} = 0 \to \sum_{i=1}^{l}\alpha_i y_i = 0 \qquad (4)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \to \alpha_i = C\xi_i \qquad (5)$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \to y_i(\varphi(x_i)\cdot w + b) - 1 + \xi_i = 0 \qquad (6)$$

Substituting (3) and (5) into (6), we can solve the resulting equation and (4) for α and b. Then, the decision function is given by

$$f(x) = \text{sgn}(\sum_{i=1}^{l}(\alpha_i y_i < \varphi(x), \varphi(x_i) >) + b)$$

For an arbitrary instance x, if $f(x) = 1$, it is classified into the positive class; otherwise, it belongs to the negative class.

The geometric interpretation of the above problem (1) with $x \in \mathbb{R}^2$ for linearly separable case is shown in Fig. 1, where minimizing $\frac{1}{2}\|w\|^2$ realizes the maximal margin between the straight lines

$$< w, x > +b = 1 \text{ and } < w, x > +b = -1 \qquad (7)$$

Minimizing $\sum_{i=1}^{l}\xi_i^2$ makes the two straight lines (7) proximal to all inputs of positive ("+") and negative ("○") examples respectively.

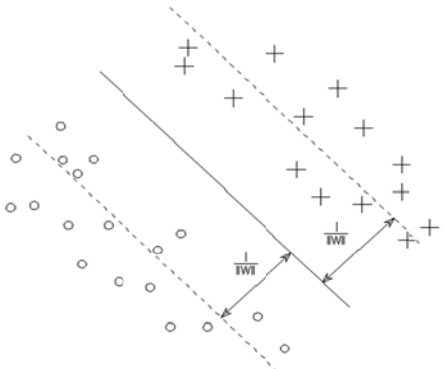

Figure 1. An interpretation of two-dimensional least squares SVM.

It's worth mentioning that the superiority of least squares SVM is simpler and faster than SVM. This is because least squares SVM needs to solve a quadratic programming with only equality constraints, or equivalently a linear system of equations, but SVM needs to solve a quadratic programming with inequality constraints.

B. Biased-SVM

In a general PU learning problem, the training dataset is represented as $T = \{(x_1, y_1), \cdots, (x_p, y_p), x_{p+1}, \cdots, x_{p+u}\}$, where $x_i \in \mathbb{R}^n, i = 1, \cdots, p+u$ are input examples, $y_i = 1, i = 1, \cdots, p$ represent the labels of input positive examples. p and u are the numbers of positive and unlabeled examples respectively, $p \ll u$. We try to find a decision function $f(x)$ to predict the class label of $\forall x \in \mathbb{R}^n$.

One of the popular methods for PU learning was proposed based on SVM technique in [16] which was called Biased-SVM. More concretely, Biased-SVM took unlabeled examples as negative examples with noise, namely $y_i = -1,\ (i = p+1, \cdots, p+u)$, and then constructed an SVM classifier by giving a larger penalty parameter to weight the positive examples errors and a smaller penalty parameter to weight the unlabeled examples errors because the unlabeled dataset, which was assumed to be negative examples dataset in Biased-SVM, also contained positive examples. Then the optimization problem of Biased-SVM could be formulated as

$$\begin{aligned} &\min_{(w,b,\xi)} \quad \frac{1}{2}\|w\|^2 + C_p\sum_{i=1}^{p}\xi_i + C_n\sum_{i=p+1}^{p+u}\xi_i \\ &s.t.\ \ y_i(< w, \varphi(x_i) > +b) \geq 1 - \xi_i \quad i = 1, 2, ..., p+u \\ &\qquad \xi_i \geq 0 \quad i = 1, 2, ..., p+u \end{aligned} \tag{8}$$

where $\xi_i \ \ i = 1, 2, ..., p+u$ are penalizing variables. C_p and C_n represent the penalty parameters of misclassification for positive and unlabeled examples respectively. Similar to classical SVM, function $\varphi(\cdot)$ maps the input space into a higher dimensional space when the dataset is nonlinearly separable. Then, the decision function is given by

$$f(x) = \text{sgn}(\sum_{i=1}^{p+u}(\alpha_i y_i < \varphi(x), \varphi(x_i) >) + b)$$

III. A Biased Least Squares Support Vector Machine For PU Learning

In this paper, we still mainly consider PU learning. From the definition of PU learning in section II, we observe that PU learning is different from classical binary classification, in which both positive and negative training examples are required. The key feature of PU learning is that there are no labeled negative training examples, which makes the traditional classification techniques inapplicable. Therefore, the key task for PU learning is how to exploit underlying information of unlabeled examples sufficiently and accurately for classification.

Similar to the idea of Biased-SVM [16], we construct a biased least squares SVM classifier by giving two different penalty parameters to weight the misclassification errors of positive and unlabeled examples respectively, where unlabeled examples can be regarded as negative examples with noise. i.e., $y_i = -1, i = p+1, \cdots, p+u$. Thus, we seek the decision function so that the classification hyper-plane separates both positive examples and unlabeled examples with the maximum margin.

$$\begin{aligned} &\min_{(w,b,\xi)} \quad \frac{1}{2}\|w\|^2 + \frac{C_p}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{C_n}{2}\sum_{i=p+1}^{p+u}\xi_i^2 \\ &s.t.\ \ y_i(< w, x_i > +b) = 1 - \xi_i \quad i = 1, 2, ..., p+u \end{aligned} \tag{9}$$

where equality constraint in optimization problem (9) and $C_p\sum_{i=1}^{p}\xi_i^2 + C_n\sum_{i=p+1}^{p+u}\xi_i^2$ imply that all positive examples approximate to straight line $< w, x > +b = 1$ and unlabeled examples approximate to $< w, x > +b = -1$ respectively. However, the degree of approximation for positive and unlabeled examples is different, which is controlled by the parameters C_p and C_n. As we all know, the larger the penalty parameter is, the more possibility the example is classified accurately. Obviously, C_p is larger than C_n because we pay more attention to classify positive examples for PU learning.

In order to solve the above optimization problem (9) more concisely, we rewrite it as matrix format

$$\begin{aligned} &\min \frac{1}{2}w^T w + \frac{1}{2}q^T C q \\ &s.t.\ D\left(X \cdot w + \text{e}b\right) + q = \text{e} \end{aligned} \tag{10}$$

where

$$C = \begin{pmatrix} C_p & 0 & \cdots & \cdots & & 0 \\ 0 & \ddots & & \ddots & & \\ \vdots & & C_p & & \ddots & \\ \vdots & \ddots & & C_n & & \\ & & \ddots & & \ddots & 0 \\ 0 & & \cdots & \cdots & 0 & C_n \end{pmatrix}_{(p+u)\times(p+u)}$$

$$D = \begin{pmatrix} y_1 & 0 & & 0 \\ 0 & y_2 & \ddots & \\ & \ddots & \ddots & 0 \\ 0 & & 0 & y_{p+u} \end{pmatrix}_{(p+u)\times(p+u)}$$

$x_1, ..., x_{p+u}$ constitute a matrix X and $q = \left(\xi_1, ..., \xi_{p+u}\right)^T$.

It is easily observed that the minimizing problem (10) is a quadratic programming with only equality constraints. Therefore, we can solve a set of linear equations. Introducing the Lagrange multiplier $\alpha = (\alpha_1, \cdots, \alpha_{p+u})^T$ from optimization problem (10), we obtain

$$L\left(w, b, q, \alpha\right) = \frac{1}{2}w^T w + \frac{1}{2}q^T C q - \alpha^T D\left(x \cdot w + \text{e}b\right) - \alpha^T q + \alpha^T \text{e} \tag{11}$$

Taking the partial derivatives of (11) with respect to w, b, q, α and equating them to zero, we obtain the optimal conditions as follows

$$\frac{\partial L}{\partial w} = w - X^T D\alpha = 0 \Rightarrow w = X^T D\alpha \tag{12}$$

$$\frac{\partial L}{\partial b} = \text{e}^T D\alpha = 0 \tag{13}$$

$$\frac{\partial L}{\partial q} = cq - \alpha \Rightarrow q = c^{-1}\alpha \tag{14}$$

$$\frac{\partial L}{\partial \alpha} = D\left(X \cdot w + eb\right) - q + e = 0 \tag{15}$$

Substituting (12) and (14) into (15), we can solve the resulting equation and (13) for α and b. Namely,

$$\left(DXX^T D + C^{-1}\right)\alpha + Deb = e \tag{16}$$

Thus, the decision function is given by

$$f(x) = \text{sgn}(g(x)) = \text{sgn}(\sum_{j=1}^{p+u} \alpha_j y_j < x, x_j > +b)$$

Like least squares SVM, we also introduce a nonlinear function $\varphi(\cdot)$ mapping the input feature space to the higher dimensional space for the nonlinearly separable case. Therefore, we need to solve the following optimization problem

$$\min_{(w,b,\xi)} \quad \frac{1}{2}\|w\|^2 + \frac{C_p}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{C_n}{2}\sum_{i=p+1}^{p+u}\xi_i^2 \tag{17}$$

$$s.t. \ y_i(< w, \varphi(x_i) > +b) = 1 - \xi_i \ \ i = 1, 2, ..., p + u$$

Similar to derivation of for linearly separable case, we can solve the following liner equations

$$\left(DKD + C^{-1}\right)\alpha + Deb = e$$

Setting kernel tricks: $K(x, x_i) =< \varphi(x), \varphi(x_i) >$ can avoid the explicit treatment of variables in the feature space. The decision function is

$$f(x) = \text{sgn}(g(x)) = \text{sgn}(\sum_{j=1}^{p+u} \alpha_j y_j K(x, x_j) + b)$$

In order to depict concisely, we call the proposed classifier as Biased-LSSVM in short.

Compared with Biased-SVM, Biased-LSSVM is more suitable for PU learning. First, Biased-LSSVM can reflect the class labels of all examples more sufficiently and accurately than Biased-SVM. This is because that only support vectors determine the final classifier for Biased-SVM, whereas the whole examples are used to construct the final classifier for Biased-LSSVM. Obviously, negative support vectors may contain some positive examples and produce some effects on the construction of Biased-SVM classifier. Contrarily, if all examples take part in the building of classifier, compared with most correct negative examples, those false negative examples are not important for the final classifier construction. Second, Biased-LSSVM is a more stable classifier than Biased-SVM. Namely, the performance of Biased-LSSVM changes less than that of Biased-SVM over a wide ratio of positive examples in unlabeled examples. This is because that the number of negative examples is far more than positive examples in unlabeled examples and the distribution of negative class changes little over a wide ratio of positive examples in unlabeled examples. Third, the time complexity of Biased-LSSVM is lower than that of Biased-SVM, where Biased-LSSVM only needs to solve liner equations and Biased-SVM is a quadratic programming.

IV. Experiments

To evaluate the performance of Biased-LSSVM, we perform experiments on two real-world applications: text classification and bioinformatics classification in this paper. More concretely, we compare Biased-LSSVM with Biased-SVM, EB-SVM, ROC-SVM and S-EM on text classification, which are the most popular methods. And we just compare Biased-LSSVM with Biased-SVM and EB-SVM on palmitoylation and phosphorylation sites prediction because most two-steps methods are unsuitable for bioinformatics classification. In both classification datasets, we use LIBSVM[1] to build a classifier for Biased-SVM. LPU package [2] is used for the implementation of S-EM, ROC-SVM for the text classification datasets.

We use the popular F score on the positive class as the evaluation measure. F score takes into account of both recall and precision

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{18}$$

where

$$precision = \frac{TP}{TP + FP} \text{ and } recall = \frac{TP}{TP + FN}$$

TP, *TN*, *FP* and *FN* are the number of true positive, true negative, false positive and false negative, respectively. A high precision ensures that the identified positives are predominantly true positives, and a high recall ensures that most of the positives are identified. F score captures the average effect of both precision and recall, and is therefore suitable for our purpose.

F score cannot be computed on the validation set during the training process because there is no negative example. An approximate computing method [17] is used to evaluate the performance by

$$\hat{F} = \frac{r_p^2}{P(f(X) = 1)} \tag{19}$$

where X is the random variable representing the input vector, $P(f(X) = 1)$ is the probability of an input example X classified as positive example, r_p is the recall for positive set P in the validation set.

A. Text Classification

- Data Preprocessing

Reuters[3] corpus are used to construct text datasets. For Reuters corpus, the top ten popular categories are used. Each category is employed as the positive class, and the rest categories are employed as the negative class. This gives us 10 datasets. In data pre-processing, we apply stop word removal, but no feature selection or stemming is done. Each document is represented as a vector of TF-IDF value. For each dataset, 30% of the documents are randomly selected as test documents. The remaining

[1] LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[2] http://www.cs.uic.edu/~liub/LPU/LPU-download.html

[3] http://www.daviddlewis.com/resources/testcollections/reuter21578/

(70%) are used to create training dataset as follows: δ percent of the documents from the positive class are first selected as the positive dataset P . The rest of the positive documents and all negative documents are mixed as unlabeled dataset U . δ is selected as 0.3 and 0.7.

In the experiment, the linear kernel function is used since it always performs excellently for text classification tasks [26]. 30 percent of training examples set constitutes the validation set. Penalty factors C_p and C_n are optimized on validation sets, which are selected from the set: $\{4^{-10}, 4^{-9}, \cdots, 4^{6}\}$.

- Experimental Results

We now present the experimental results. Table I shows the classification results of various techniques in terms of F score on $\delta = 0.3$ on 10 datasets. Due to the idea of Biased-LSSVM is similar to Biased-SVM, we first compare Biased-LSSVM with Biased-SVM. It is obvious that F scores of Biased-LLSVM are higher than that of Biased-SVM on the fourth dataset and the last four datasets. The final row of the Table I give the average results of each column. Average F score of Biased-LSSVM equals to 0.8310 and is far greater than Biased-SVM. In addition to compare with Biased-SVM, Biased-LSSVM also compares with S-EM [6], ROC-SVM [8] and EB-SVM [18]. We observe that Biased-LSSVM produces better results than S-EB and ROC-SVM, but slightly worse than EB-SVM.

TABLE I.
F SCORES COMPARISON BETWEEN BIASED-LSSVM AND OTHER METHODS ON $\delta = 0.3$ ON REUTERS

Positive class set	F score				
	Biased-LSSVM	Biased-SVM	EB-SVM	ROC-SVM	S-EM
1 earn	0.9498	0.9591	0.9803	0.8768	0.9457
2 acq	0.8954	0.8970	0.9393	0.9157	0.9342
3 crude	0.7516	0.8140	0.8571	0.8846	0.8571
4 trade	0.8280	0.7795	0.8824	0.8519	0.7980
5 money-fx	0.7297	0.7925	0.7597	0.7817	0.7485
6 interest	0.8034	0.8142	0.8361	0.8625	0.8309
7 ship	0.6452	0.5924	0.5763	0.5570	0.5502
8 sugar	0.9254	0.7636	0.9552	0.8303	0.7050
9 coffee	0.9063	0.8022	0.9375	0.8706	0.8155
10 gold	0.8750	0.7495	0.7826	0.7101	0.7818
Average	0.8310	0.7964	0.8507	0.8141	0.7967

Similar to Table I, Table II shows F scores comparison between Biased-LSSVM and other methods on $\delta = 0.7$ on Reuters. From Table II, we observe that Biased-SVM and Biased-LSSVM have little difference in average F score. But Biased-LSSVM is still higher than ROC-SVM and S-EM, lower than EB-SVM in average F score.

On the other hand, compared with the results in Table I, the average F score of all methods in Table II grows. In other words, the performance of all these methods increases when the percentage of known positives changes from 0.3 to 0.7.

TABLE II.
F SCORES COMPARISON BETWEEN BIASED-LSSVM AND OTHER METHODS ON $\delta = 0.7$ ON REUTERS

Positive class set	F score				
	Biased-LSSVM	Biased-SVM	EB-SVM	ROC-SVM	S-EM
1 earn	0.9584	0.981	0.9842	0.9594	0.9473
2 acq	0.9251	0.9525	0.9572	0.8696	0.9223
3 crude	0.9474	0.904	0.9348	0.9198	0.8989
4 trade	0.8636	0.9143	0.9195	0.9098	0.7659
5 money-fx	0.7285	0.8633	0.8633	0.8786	0.6469
6 interest	0.8739	0.8547	0.8547	0.8662	0.7238
7 ship	0.8462	0.7945	0.8462	0.8474	0.7474
8 sugar	0.9231	0.9565	0.9565	0.8886	0.8985
9 coffee	0.9697	0.9375	0.9851	0.8861	0.8896
10 gold	0.8772	0.9091	0.9091	0.7887	0.7627
Average	0.8913	0.9067	0.9211	0.8814	0.8203

B. *Bioinformatics Classification*

In this experiment, we use two datasets, the prediction of palmitoylation and phosphorylation sites in proteins. In next subsection, we will introduce these two datasets.

- Backgrounds of Bioinformatics

Protein palmitoylation plays important roles in cell signaling associated with cellular dynamics and plasticity. It regulates epidermal homeostatic and hair follicle differentiation [27], and plays a key role in neuronal development and synaptic plasticity[28].

Phosphorylation is involved in diverse signal transduction pathways. Most approaches are based on traditional supervised learning. i.e., they used the non-annotated sites of the phosphoBase as negative sites. But in fact, it contains many more false negative sites than phosphoBase. Thus, it is more reasonable that we regard non-annotated sites of the phosphoBase as negative sites. In this paper, we mainly concern kinase family: cyclin-dependent kinase.

- Data preparing

For the S-palmitoylation sites, the dataset used in this research comes from [29]. We get 344 experimentally verified palmitoylation sites as our final positive dataset and 1815 cysteine sites as unlabeled dataset. We call this PU dataset as Palm in short.

For the phosphorylation sites, the dataset comes from [30]. We only use binary encode scheme which its window length are set to 11 and 12 for CDK, named CDK and CDK2. Information of these datasets is summarized in Table III.

TABLE III.
DATASETS USED IN THE EXPERIMENTS FOR PU LEARNING

Datasets	# instance	# Class	# Feature
CDK	2446	2	276
CDK2	1830	2	276
Palm	2159	2	120

- Experiment Preprocessing

In this experiment, radial basis function (RBF) kernel is used for the prediction of palmitoylation and phosphorylation sites in proteins. It can be written as

$$K(x_i, x_j) = \exp(-\left\| x_i - x_j \right\|^2 / \sigma^2)$$

where x_i and x_j are examples, σ is kernel parameter selected from $\{4^{-10}, 4^{-9}, \cdots, 4^{6}\}$. Penalty parameters of C_p and C_n are selected from: $\{4^{-10}, 4^{-9}, \cdots, 4^{6}\}$ and $C_p \ge C_n$. All parameters are optimized by a ten-fold cross validation procedure on the training dataset which give the best F-score for each method.

In order to create a wide range of scenarios, we select δ from 10% to 90% (0.1-0.9) examples from experimentally verified palmitoylation sites randomly as positive dataset P. The rest of the positive examples and all 1815 cysteine sites are mixed as unlabeled dataset U. For each δ, we access the performance of the classifier. In addition, for the prediction on phosphorylation sites, we just select δ as 0.3 and 0.7 to present the classification results.

- Experimental Results

Fig. 2 and Fig. 3 show the mean of precision and recall at different percentage δ for the prediction of palmitoylation sites in proteins. We observe from Fig. 2 that the precision of Biased-SVM decreases with the increase of δ. Contrarily, Biased-LSSVM and EB-SVM increase with the increase of δ. From Fig. 3, it is obvious the recall of Biased-LSSVM is much higher than Biased-SVM and EB-SVM whatever δ is. Both Fig. 2 and Fig. 3 show that Biased-LSSVM classifies less false negative examples than Biased-SVM and EB-SVM, although it classifies less true positive examples than Biased-SVM and EB-SVM from unlabeled examples at the same time.

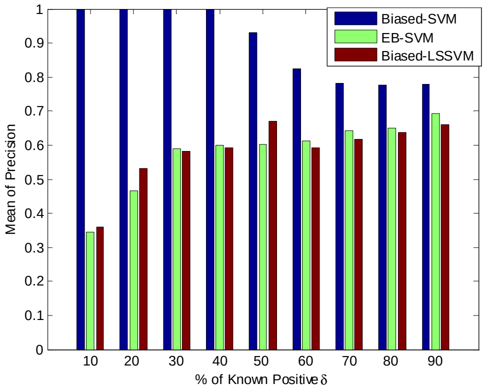

Figure 2. Average precision comparison Biased-LSSVM with Biased-SVM and EB-SVM on the prediction of palmitoylation sites.

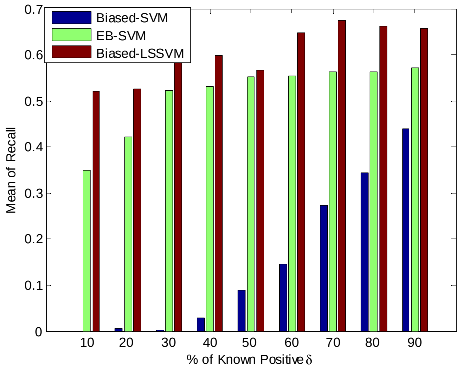

Figure 3. Average recall comparison Biased-LSSVM with Biased-SVM and EB-SVM on the prediction of palmitoylation sites.

Fig. 4 shows the combination of precision and recall performance by the average F score. It can be noticed that Biased-LSSVM, Biased-SVM and EB-SVM exhibit a progressively increment in performance when δ grows from 10% to 90%. Biased-LSSVM possesses of the highest F score than Biased-SVM and EB-SVM on every different δ on the prediction of palmitoylation sites in proteins, especially when δ is smaller. In a word, the performance of Biased-LSSVM is the most effective on the prediction of palmitoylation sites.

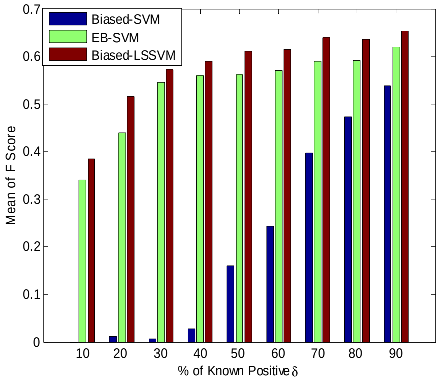

Figure 4. Average F score comparison Biased-LSSVM with Biased-SVM and EB-SVM on the prediction of palmitoylation sites.

In order to more visualized, Average precision, recall and F score of Biased-LLSVM and Biased-SVM are shown on different percentage δ on the prediction of palmitoylation sites in proteins in Table IV and Table V. Biased-LLSVM produces the best results consistently cross all varied δ.

TABLE IV.
AVERAGE PRECISION AND RECALL COMPARISON BIASED-LSSVM WITH BIASED-SVM AND EB-SVM ON THE PREDICTION OF PALMITOYLATION SITES

δ	Average Precision			Average Recall		
	Biased-LSSVM	Biased-SVM	EB-SVM	Biased-LSSVM	Biased-SVM	EB-SVM
0.1	0.3589	1.0000	0.3454	0.5208	0.0000	0.3487
0.2	0.5321	1.0000	0.4667	0.5256	0.0059	0.4213
0.3	0.5819	1.0000	0.5896	0.6010	0.0029	0.5229
0.4	0.5932	1.0000	0.5991	0.5986	0.0292	0.5318
0.5	0.6717	0.9300	0.6032	0.5670	0.0900	0.5533
0.6	0.5931	0.8235	0.6120	0.6484	0.1455	0.5547
0.7	0.6169	0.7810	0.6427	0.6740	0.2734	0.5629
0.8	0.6367	0.7753	0.6503	0.6626	0.3431	0.5638
0.9	0.6598	0.7782	0.6936	0.6572	0.4392	0.5723

TABLE V.
AVERAGE F SCORE COMPARISON BIASED-LSSVM WITH BIASED-SVM AND EB-SVM ON THE PREDICTION OF PALMITOYLATION SITES

δ	Average F Score		
	Biased-LSSVM	Biased-SVM	EB-SVM
0.1	0.3843	0.0000	0.3409
0.2	0.5154	0.0111	0.4401
0.3	0.5720	0.0057	0.5445
0.4	0.5891	0.0281	0.5599
0.5	0.6107	0.1603	0.5616
0.6	0.6138	0.2434	0.5693
0.7	0.6384	0.3968	0.5901
0.8	0.6361	0.4734	0.5910
0.9	0.6532	0.5386	0.6188

For the prediction of phosphorylation sites, numerical results are presented in Table VI. The results shown in boldface are significantly better than others. Obviously, Biased-LSSVM works much better than Biased-SVM and EB-SVM on CDK and CDK2 because it focuses on effectively extracting the positive examples from unlabeled examples set, so that the resulting classifier is near the optimal positive and negative boundary.

TABLE VI.
AVERAGE F SCORES COMPARISON BIASED-LSSVM WITH BIASED-SVM AND EB-SVM ON $\delta = 0.3$ AND $\delta = 0.7$ ON CDK AND CDK2

Methods	CDK2		CDK	
	$\delta = 0.3$	$\delta = 0.7$	$\delta = 0.3$	$\delta = 0.7$
Biased-LSSVM	**0.4257**	**0.431**	**0.496**	**0.5053**
Biased-SVM	0.408	0.376	0.446	0.415
EB-SVM	0.3515	0.391	0.45	0.4613

In addition, time complexity with respect to Biased-LSSVM and Biased-SVM is shown in Fig 5. on the prediction of phosphorylation sites. It is reasonable to compare the time complexity of Biased-LSSVM and Biased-SVM on CDK and CDK2 because the number of features of CDK is equal to CDK2. It is obvious from Figure 5 that the time complexity of Biased-LSSVM is lower than Biased-SVM whatever the dataset is. Moreover, the run time of Biased-SVM increases higher than that of Biased-LSSVM with the number of examples increasing (from CDK2 to CDK). This is because the time complexity of Biased-LSSVM is linear, whereas that of Biased-SVM is quadratic with respect to the number of training examples.

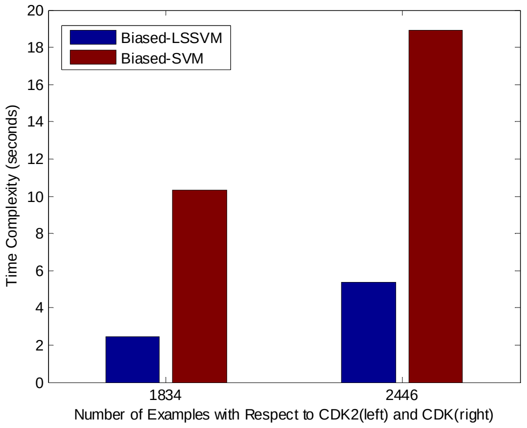

Figure 5. Time complexity with respect to Biased-LSSVM and Biased-SVM on the prediction of phosphorylation sites.

V. CONLUSIONS

In this paper, we have put forward a biased least squares SVM classifier for PU learning, named Biased-LSSVM. The proposed classifier is constructed by giving two different penalty parameters to classification errors of positive examples and unlabeled examples which are regarded as negative examples with noise. Experimental results on two different applications have shown that Biased-LSSVM is more effective than Biased-SVM and other popular methods for PU learning. Namely, Biased-LSSVM has more strong discriminative power for positive and unlabeled learning.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (11371365, 11071252), Chinese Universities Scientific Fund (2012YJ130, 2013YJ010). The authors would like to thank Yuxin Li for having provided us with her bioinformatics dataset.

REFERENCES

[1] Y. Jin, C.W. Huang, L. Zhao, “A Semi-Supervised Learning Algorithm Based on Modified Self-training SVM” , Journal of Computers, 2011, 6(7).

[2] Z.Y. Yao, “Semi-Supervised MEC Clustering Algorithm on Maximized Central Distance”, Journal of Computers, 2013, 8(10).

[3] X.D. Xu, “A New Sub-topics Clustering Method Based on Semi-supervised Learning”, Journal of Computers, 2012, 7(10), Special Issue: Advances in Information and Computers.

[4] F. Denis, “PAC learning from positive statistical queries”, Lecture Notes in Computer Science, 1501, 1998, pp. 112-126.

[5] S. Muggleton, “Learning from the positive data, Machine Learning”, Inductive Logic Programming, Lecture Notes in Computer Science, 1314 1997): 358-376.

[6] B. Liu, W.S. Lee, P.S. Yu, et al, “Partially supervised classification of text documents”, In: Proceedings of the 19th International Conference on Machine Learning. 2002, pp. 387-394.

[7] H. Yu, J. Han, K.C.C. Chang, “PEBL: Web Page Classification without Negative Examples”, IEEE Transactions on Knowledge and Data Engineering. 16(1), 2004, pp. 70-81.

[8] X.L. Li, B. Liu, “Learning to Classify Text Using Positive and Unlabeled Data”, In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico. 2003, pp. 587–594.

[9] G.P.C. Fung, J.X. Yu, H. Lu, P.S. Yu, Text Classification without Negative Examples Revisit, IEEE Transactions on Knowledge and Data Engineering 18(1) , 2006, pp. 6-20.

[10] M.N. Nguyen, X.L. Li, S.K. Ng, “Positive Unlabeled Learning for Time Series Classification”, In: Proceedings of International Joint Conference on Artificial Intelligence, IJCAI , 2011.

[11] C.X. Zhu, B. Liu, Q.Z. Yu, X.Z. Liu, W.X. Yu, “A Spy Positive and Unlabeled Learning Classifier and Its Application in HR SAR Image Scene Interpretation”, IEEE Radar Conference (RADAR), 2012, pp. 0516 – 0521.

[12] M.N. Nguyen, X.L. Li, S.K. Ng, “Ensemble Based Positive Unlabeled Learning for Time Series Classification”, LNCS. 7238, 2012, pp. 243–257.

[13] D. Zhang, W.S. Lee, “A simple probabilistic approach to learning from positive and unlabeled Examples”, In Proceedings of the 5th Annual UK Workshop on Computational Intelligence (UKCI). 2005, pp. 83-87.

[14] C. Elkan, K. Noto, “Learning Classifiers from Only Positive and Unlabeled Data”, In Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA. 2008, pp. 213-220.

[15] C. Luigi, E. Charles, C. Michele, “Learning gene regulatory networks from only positive and unlabeled data”, BMC Bioinformatics. 11(1): 2010, pp. 228.

[16] B. Liu, Y. Dai, X.L. Li, W.S. Lee, P.S. Yu, “Building Text Classifiers Using Positive and Unlabeled Examples”, In Proceedings of the 3rd IEEE International Conference on Data Mining, Melbourne, Florida, United States. 2003, pp.179-188.

[17] W.S. Lee, B. Liu, “Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression”, In Proceedings of the 20th International Conference on Machine Learning, Washington DC, United States. 2003, pp. 448-455.

[18] T. Ke, B. Yang, J. Tan, J. Ling. “Building High-performance Classifiers on Positive and Unlabeled Examples for Text Classification”, Advances in Neural Networks – ISNN 2012,Lecture Notes in Computer ScienceVolume 7368, 2012, pp. 187-195.

[19] F. Letouzey, F. Denis, R. Gilleron, E. Grappa, “Learning from positive and unlabeled examples”, ALT. 2000). Algorithmic Learning Theory Lecture Notes in Computer Science Volume 1968, 2000, pp.71-85.

[20] F. Denis, R. Gilleron, F. Letouzey, “Learning from Positive and Unlabeled Examples”, Theoretical Computer Science 348(1): 2005, pp. 70-83.

[21] Y. Xiao, M.R. Segal, “Biological sequence classification utilizing positive and unlabeled data”, Bioinformatics. 24: 2008, pp. 1198–1205.

[22] M. Fantine, V. Jean-Philippe, “A bagging SVM to learn from positive and unlabeled Examples”, Technical Report hal-00523336, version 1, HAL. 2010.

[23] G. Alireza, R.R. Hamid, F. Mohsen, T.M. Mohammad, H.R. Mohammad, “Active Learning from Positive and Unlabeled Data”, 11th IEEE International Conference on Data Mining Workshops. 2011, pp. 244-250.

[24] J. A. K. Suykens, “Least squares support vector machines for classification and nonlinear modeling”, Neural Network World, 10(1–2): 2000, pp. 29–47.

[25] Cortes, C., Vapnik, V.: “Support vector network”, J. Machine. Learning. 20, 1995, pp. 273-297.

[26] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, In: Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 137-142.

[27] P. Mill, A.W. Lee, Y. Fukata, R. Tsutsumi, M. Fukata, M. Keighren, R.M. Porter, L. McKie, L. Smyth, I.J. Jackson, “Palmitoylation regulates epidermal homeostasis and hair follicle differentiation”, PLOS Genetics. 5(11): 2009. 1-15 e1000748.

[28] Y. Fukata, M. Fukata, “Protein palmitoylation in neuronal development and synaptic plasticity”, Nat. Rev. Neurosci. 11: 2010, pp. 161-175.

[29] Y.X. Li, Y.H. Shao, N.Y. Deng, “Improved Prediction of Palmitoylation Sites Using PWMs and SVM”, Protein & Peptide Letters 18: 2011, pp. 186–193.

[30] Z.M. Yin, J.Y. Tan, “New encoding schemes for prediction of protein Phosphorylation sites”, IEEE 6th International Conference on Systems Biology (ISB). 2012, pp.55-62.

Ting Ke received the B.S. degree and Master degree in mathematics from Inner Mongolia University and Henan Polytechnic University, China, in 2005 and 2008, respectively. Shi is a Ph. D. student at China Agricultural University, Beijing, China. Her main research interesting includes *semi-supervised learning, PU learning and optimization research and applications.*

Lujia Song is a M. S. candidate of China Agricultural University, Beijing, China. Her main research interests lie in *data mining models and application*

Bing Yang is a Ph. D. candidate of China Agricultural University, Beijing, China. His main research interests lie in *data mining models and applications.*

Xinbin Zhao received the B.S. degree and Master degree in College of Applied Sciences from Beijing University of Technology, Beijing, China, in 2005 and 2009. He is a Ph.D. candidate of China Agricultural University, Beijing, China. His current research interests include *machine learning, pattern recognition and their applications in the field of computer vision.*

Ling Jing received Master's degree and Ph. D. in mathematics from Chongqing University and Beijing University of Aeronautics & Astronautics, China, in 1994 and 1997, respectively. She is a professor and Ph. D. supervisor a China Agricultural University. Her main research interests lie in *data mining and optimization researc*