Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 09-10/8/2018
DOI: 10.15625/vap.2018.0003

AN IMPROVED SVM METHOD FOR IMBALANCED DATA AND ITS APPLICATION IN LINK PREDICTION IN CO-AUTHORSHIP NETWORKS

Trinh Khac Linh, Tran Dinh Khang, Pham Minh Chuan
Hanoi University of Science and Technology
linhtk.dhbk@gmail.com, khangtd@soict.hust.edu.vn

ABSTRACT: *In classification problems, the class imbalance significantly affectsthe efficiency of classification models. There are several proposals on improving SVM methods to adapt to imbalanced data sets. This paper proposes an improved SVM method for imbalanced data through adjusting weighted vector w, while combining with the Weighted-SVM training method, to increase the efficiency of classification for imbalanced data and apply to link prediction problem in co-authorship networks.*

Keywords: *co-authorship networks, link prediction, imbalanced classification, support vector machine.*

I. INTRODUCTION

In classification domain, imbalanced dataset occurs when instance of one class outnumbers the instance of other class. The class which overwhelms is called the majority class while the other is called minority class. This is one of the challenging problems in data mining research that degrades the performance of each classification method. Almost all classifiers such as decision tree, support vector machine (SVM) are designed for the accuracy but it does not look at any class. As rare instances occur infrequently, classification rules that predicting the small classes tends to be rare, undiscovered or ignored; consequently, test samples belonging to the small classes are misclassified more often than those belonging to the prevalent classes. For example, a training set has one million records, but only 10 instances of rare class. A learning method has reached an accuracy of 99.98% but predicts completely misclassified the minority class.

Support vector machine (SVM) is a classification model which widely applies to many real-world classification problems from various domains. SVM learns a hyperplane $f(x) = w.x + b$ with widest margin saparating the two classes. Maximizing the margin is formalized as a convex quadratic programming problem. SVM uses only a set of support vectors to construct classification models and focuses on maximizing the margin between examples of opposite classes with a penalty for errors. For imbalanced training data, the separating hyperplane learned by the SVM is very close to the minority class, which leads to low performance or no generalization at all instances from this class [9]. Some strategies based on algorithm-level have been proposed to improve the performance of SVM on imbalanced datasets. In the Weighted-SVM method [2], the SVM objective function is modified to assign different misclassification costs, C^+, C^-, instead of the same cost (i.e. C) for both positive and negative misclassification in the penalty term in the standard SVM. z-SVM [3] adjusted the weight vector w in the decision function of the standard SVM trained model, to obtain a good margin of separation for the positive class.

Co-authorship network or academic social network is a typical social network that can be formularized by a graph in which a node is an author or researcher and an edge reflects the connection between them in terms of having joint paper(s). Co-authorship network contains abundant academic characters in comparison with other social networks, so that analysing and mining information from co-authorship network have significant and practical meanings in modeling and increasing research quality [13]. Link prediction in co-authorship network is one of the important problems in social network research. Researchers have focused on analyzing and proposing solutions to give efficient recommendation to authors who can work together in a science project (e.g. a paper). Link prediction in the co-authorship network strengthens collaboration and idea exchange between scientists. The aim of link prediction is to determine couples of authors who can collaborate in the future based on some features of the current network structure such as similarity measures between nodes, information of authors, publish papers, etc. A feature vector $f_{ij} \in F$ consists of several attributes, computed for the node pair (v_i, v_j). A predictor $p : F \rightarrow \{true, false\}$ is a function that maps feature vectors to the binary space. A good predictor is one in which $p(f_{ij}) = y_{ij}$ holds true for a large proportion of the test feature vectors $f_{ij} \in Fs$. We build predictors by training a learning algorithm to generate the model on the training set Fr. Co-authorship network is a typical network with a large number of candidate pairs, however, the number of real linked pairs is very small that may cause class imbalance and decrease in performance of link prediction methods.

In this paper, we propose an improved SVM method through adjusting weighted vector w, while combining with the Weighted-SVM training method to adapt to imbalanced datasets, and appling for link prediction problem in co-authorship network. The rest of the paper is organized as follows. Section 2 reviews the SVM method and its existing improvements. Section 3 proposes the improved SVM for imbalanced datasets. The link prediction problem and experimental results on the imbalanced datasets are presented in Section 4 and Section 5. Finally, conclusions are delineated in Section 6.

II. SVM ON IMBALANCED DATASETS

A. Standard SVM[1,2,8]

Support vector machine (SVM) is a classification model which is widely applied to many real-world classification problems of various domains. SVM learns a hyperplane $f(x) = w.x + b$ with maximum margin separating hyperplane from two classes based on two support hyperplanes from each class. Maximizing the margin is formalized as a convex quadratic programming problem. If the training dataset is linearly separable with the exception of some outliers, by using the slack variable ξ_i, a classifier can learn by maximizing the soft margin which is formalized as $\frac{2}{||w||}$. For a binary classification task on a set of training data: N input vectors $\{x_1, x_2, \dots, x_N\}$ and it's label $\{y_1, \dots, y_N\}$, where $y_i \in \{-1, +1\}$ is the label associated with x_i. SVM solves the following primal optimization problem:

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

Subject to: $\begin{cases} y_i(w.x_i + b) + \xi_i \geq 1, i = 1, \dots, N \\ \xi_i \geq 0 \end{cases}$ (2)

Where C > 0 is the regularization parameter, which is a trade-off between maximization of the margin and minimization of the training errors. When the value of C is large, the optimization will choose a small margin hyperplane. When the value of C is small, the optimization will choose a large margin separating hyperplane. By applying Lagrange duality and kernel methods, the following dual optimization problem is obtained from (1):

$$\max_{\alpha_i \in R} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \quad (3)$$

Subject to: $\begin{cases} 0 \leq \alpha_i \leq C, i = 1, \dots, N \\ \sum_i^N \alpha_i y_i = 0 \end{cases}$ (4)

Where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N), \alpha_1$ is Lagrangian multipliers is associated with data (x_i, y_i); which should satisfied the following Karush Kuhn-Tucker (KKT) conditions: $\alpha_i(y_i(w.\phi(x_i) + b) - 1 + \xi_i = 0$ and $(C - \alpha_i)\xi_i = 0\ i = 1, \dots, N$.

K(,) is a kernel function which maps feature vectors onto higher dimensions using map function: $x \to \phi(x) \in R^m$.
$K(x, x_i) = \phi(x)^T\phi(x_i)$. (5)

Some common kernels used with SVM:

- Linear: $K(x, z) = x^T z$
- Polynomial: $K(x, z) = (r + x^T z)^d$ where r is a free parameter, $d \in Z$
- Gaussian Radial Basic Function (RBF) Kernel: $K(x, z) = \exp\left(-\gamma\left|\left|x - z\right|\right|^2\right), \gamma > 0$
- Sigmoid: $K(x, z) = tank(x^T z + r)$

To solve the convex quadratic programming problem (3), we can apply SMO method to determine α. In SMO method, SMO finds a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem, then picks a second multiplier α_2 and optimize the pair (α_1, α_2). Using the results of SMO, we can calculate optimal weight vector w from α as presented below:

$$w = \sum_{i \in SV}\alpha_i y_i x_i \quad (6)$$

The decision function $f(x)$ can be presented as: $f(x) = \sum_i^N \alpha_i y_i K(x_i, x) + b$ (7)

SVM Algorithm:

Input: Training data sets $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_N)\}$ where $x_i, i = 1, \dots, N$ are the features, $y_i, i = 1, \dots, N$ are the label associated with x_i

Output: The solved decision function $f(x)$

Step 1: Choose a kernel $K(x, x')$ choose $C > 0$

Step 2: Solve convex quadratic programming problem to obtain $\alpha^* = [\alpha_1^*\alpha_2^* \dots \alpha_N^*]$ using SMO method[5]:

Step 3: Compute bias b^*. Choose an α_i^*:$\alpha_i^* \in (0, C)$ associated (x_i, y_i) is referred to as a support vector, then calculate bias:

$$b^* = \frac{1}{SV}\sum_i^{SV}(y_i - \sum_{i=1}^{N} y_i\alpha_i^* K(x_i, x_j)) \quad (8)$$

Step 4: Using α^* and b^*above to substitute in (7) to construct decision function $f(x)$

B. Some research in SVM-based for imbalanced datasets

SVM uses only a set of support vectors to construct classification models and focuses on maximizing the margin between the examples of opposite classes with a penalty for each error. For imbalanced training data, the separating

hyperplane learned by the SVM is very close to the minority class, leading to low performance or no generalization at all for examples from this class. Therefore, SVM performs poorly on imbalanced datasets [9]. Some strategies based on algorithm-level have been proposed to improve the performance of SVM on imbalanced datasets.

B.1. Weighted SVM[2]

The main weakness of the SVM algorithm is that the objective function given in (1) assigns the same cost (i.e., C) for both positive and negative misclassifications in the penalty term. This would cause the separating hyperplane to be skewed towards the minority class, which would finally yield a suboptimal model. Weighted SVM is proposed in [2] to overcome the same cost (i.e. C) for both positive and negative misclassifications in the penalty term. In this method, the SVM objective function is modified to assign different misclassification costs, C^+, C^- for positive and negative class respectively. The equation of prima optimization (1) is then:

$$\min_{w,b,\xi} \frac{1}{2}\left|\left|w\right|\right|^2 + C^+ \sum_{y_i=1} \xi_i + C^- \sum_{y_i=-1} \xi_i \tag{9}$$

$$\text{Subject to: } \begin{cases} y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, i = 1, \dots, N \\ \xi_i \geq 0 \end{cases} \tag{10}$$

[15] reported that if the setting the $\frac{C^-}{C^+}$ is equal to the ratio of the minority to the majority class ratio, the classification results are better. Different misclassification costs can adjust the soft margin from the minority class to the majority class, thus being more effective for imbalanced dataset problems.

B.2. z-SVM[3]

The z-SVM is a modification algorithm proposed for SVMs in [3] to learn from imbalanced datasets. In this method, firstly a standard SVM model is trained, then its decision boundary is modified to remove its bias towards the majority class. z-SVM adjusts positive support vector by multiplying all of them by with a particular value of z. Then weight vector w in (6) is then:

$$w = z * \sum_{i \in SV^+} \alpha_i y_i x_i + \sum_{i \in SV^-} \alpha_i y_i x_i. \tag{11}$$

The optimal value z^* is solved by gradually increasing the value of z from 0 to a positive value, M, and G-mean is adopted as the evaluation measure to determine the z^*. To search the optimal value z^* effectively from 0 to M, this method uses the Golden section search algorithm [11].

The experiments to compare z-SVM with standard SVM and over-sampling SMOTE_SVM method use 5 imbalanced datasets from UCI sources[12]. These experiments uses G-mean and sensitivity metrics to compare the performances of those 3 methods. From the results of experiments, z-SVM perform better than standard SVM and SMOTE-SVM method.

z-SVM adjust the weight vector w of the decision function to obtain a good margin of the separation for the positive class. However, because the position, role, and significance of each support vector are different, assigning the same value to each positive support vector cannot achieve the desired effect in improving SVM.

B.3. New bias SVM[4]

This method improves the standard SVM by adjusting the bias value. In this method, firstly the standard SVM model is trained, then the bias of decision boundary is modified by considering the number of patterns in the minority and majority classes (or considering the number of suppors vectors for the minority and majority classes) to calculate the bias value. The two improvements of the bias can be represented as below:

$$b_p = \frac{N^+\gamma + N^-\mu}{N^+ + N^-} \tag{12}$$

$$b_{p1} = \frac{N_{SV1}\gamma + N_{SV2}\mu}{N_{SV1} + N_{SV2}} \tag{13}$$

Where $\gamma = \max_{x_k \in SV^+} \sum_{i=1}^{N} \alpha_i K(x_i, x_k)$,$\mu = \min_{x_k \in SV^-} \sum_{i=1}^{N} \alpha_i K(x_i, x_k)$, N^+, N^- are the numbers of patterns in the minority and majority classes respectively, N_{SV1}, N_{SV2} are the numbers of supports vector for the minority and majority classes respectively.

The experiments to evaluate this method use 34 datasets from UCI source with imbalanced ratio from high to low. The accuracy, G-mean, sensitivity metrics of these experiments show that this method perform better than Weighted-SVM, SMOTE-SVM in some datasets.

C. Evaluation Measures[7]

A confusion matrix is a table that is often used to describe the performance of a classification model. Typical confusion matrix can be represented as below:

	Predicted		
Actual		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

where: - TP (true positive) – the instance is positive and it is predicted as positive.

- FN (false negative) - the instance is positive and it is predicted as negative.
- FP (false positive) – the instance is negative and it is predicted as positive.
- TN (true negative) – the instance is negative and it is predicted as negative.

Many metrics have been used for the assessment of the performance of classifiers. All of them are based on the four above measures. Based on the confusion matrix, the following measures are usually used to evaluate the performance of the classification algorithms:

- Precision: is defined as the ratio of the accurately predicted positive (TP) to the predicted positive (TP and FP):$Precision = \frac{TP}{TP+FP}$

- Recall: is defined as the ratio of the accurately predicted positive (TP) to the actual positive (TP and FN): $Recall = \frac{TP}{TP+FN}$. Recall and specificity are used to monitor the classification performance on each individual class

- Positive accuracy ACC^+ is defined as the ratio of the accurately predicted positive (TP) to the actual positive (TP and FN): $ACC^+ = TP/(TP + FN)$.

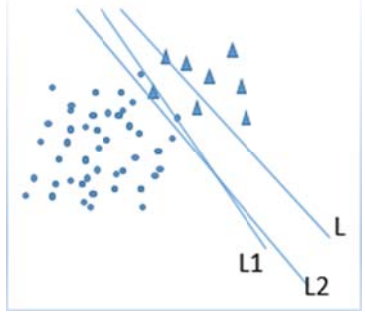
- Negative accuracy ACC^- is defined as the ratio of the accurately predicted negative (TN) to the actual negative (TN and FP): $ACC^- = TN/(TN + FP)$.

ACC^+ and $ACC^-$are used to monitor the classification performance on each individual class; however, they are not appropriate metrics when the class sizes are considerably different. In the evaluation of the classifiers, the high values of ACC^+ and ACC^- are desirable, but typically there is a tradeoff between the two. Two alternative metrics, G-mean (geometric mean) and F1-score, used to evaluate the for performance of class-imbalanced classifiers, where $F1 - score = \frac{2*Precision}{Precision+Recall}$, $G - mean = \sqrt{ACC^+ \times ACC^-}$. G-mean is the measure of the ability of a classifier to balance sensitivity and specificity.F-score is the weighted harmonic mean of the recall and precision.

III. PROPOSED IMPROVED SVM FOR IMBALANCED DATASETS

Some methods in section B adjust the cost function of each class of the samples, or adjust the classification boundary by changing weight vector or bias values. To be more specific, weighted-SVM assigns different misclassification cost for each class of the samples. However, it's difficult to determine the suitable misclassification cost for each class of the samples in practice. z-SVM assigns a weight value z to each positive support vector in the decision function. However, z-SVM seems to be hard to combine with Weighted-SVM because this may lead to changes in the ratio of Lagrange multipliers of positive to negative support vectors. Adjusting weight vector w might be more effective than adjusting bias b because adjusting w may change the values of Lagrange multiplier α or bias b.

We propose an improved SVM method through adjusting weight vector w, while combining with the Weighted-SVM training method to scope of imbalanced datasets. Firstly Weighted-SVM model is trained, then the decision boundary of the resulted model is modified by adjusting weight vector w. Unlike z-SVM, we adjusts w by increasing a particular value τ to all Lagrange multipliers of positive support vectors. This τ value adjusts a very small change of the ratio of Lagrange multipliers of positive to negative support vectors, thus improving effectively Weighted-SVM. As illustrated in right Figure, the original classification hyperplane (denoted as L) is solved by Weighted-SVM. In addition, a new hyperplane (L1) is obtained after adjusting w using τ. Adjusting w may lead to changes in b, so another new hyperplane (L2) is obtained.



The modified weight vector from (6) can be represented as follow:

$$w = \sum_{i\in SV^+}(\alpha_i + \tau)y_i x_i + \sum_{i\in SV^-}\alpha_i y_i x_i. \quad (14)$$

Also, the modified decision function of (7) can be represented as follows:

$$f(x,\tau) = \sum_{x_p\in SV^+, y_p>0}(\alpha_p + \tau)y_p K(x, x_p) + \sum_{x_n\in SV^-, y_n<0}\alpha_n y_n K(x, x_n) + b \quad (15)$$

We focus on finding the value of τ in order to achieve the the position of the hyperplane in which the value of geometric mean (G-mean) is maximum. The initial value of G-mean is obtained from Weighted-SVM on the training data sets. In each step of updating τ, G-mean value must be re-calculated on the training datasets to determine maximum G-mean and update optimal value τ^*. We setup to find the optimal value τ^* on small range $[m,M]$ such that

it is possible to obtain a small change in the ratio of Lagrange multipliers of positive to negative support vectors.There is an univariate unconstrained optimization problem, so we can apply a method based on Golden section search algorithm for the optimization process.

Algorithm:

Input: Training datasets $T = \{(x_1, y_1), (x_2, y_2), \dots (x_l, y_l)\}$ $where$ $x_i \in R^n, y \in \{1, -1\}, i = 1,2, \dots, N$

Output: Optimal value τ^*, decision function $f(x, \tau^*)$

Step 1: Learn a decision function on T by Weighted-SVM, this function can be represented by (w, b), the main output of learning is a set of support vectors SVs

Step 2: Solve the optimization problem on training set T based on Golden section search to obtain optimal value τ^*:

```
INPUT:
      Training dataset xTrain with label yTrain
      Support vectors set: αSV,xSV,ySV
OUTPUT: optimal values τ*, g*
PROCEDURE:
Initialize τ^0  τ* = τ^0,
            a = m, b = M, k, ε
Obtain: g^0 = Gmean(τ^0),  g* = g^0,
      where Gmean(τ) is a function to calculate g-mean value on training
      dataset by decision function (16) with parameter τ
WHILE b − a > ε
      τ_1^k = b − k ∗ (b − a)
      τ_2^k = a + k(b − a)
      Calculate g^1 = Gmean(τ_1^k) and g^2 = Gmean(τ_2^k)
      IF g^1 > g^2
            τ* ∈ [a, τ_2^k], b = τ_2^k
            IF g^1 > g*
                  g* = g^1, τ* = τ_1^k
            END IF
      ELSE
            τ* ∈ [τ_1^k, b], a = τ
            IF g^2 > g*
                  g* = g^2, τ* = τ_2^k
            END IF
      END IF
END WHILE
```

Step 3: Obtain the improved decision function with optimal value τ^*:

$$f(x, \tau^*) = \sum_{x_p \in SV^+, y_p > 0} (\alpha_p + \tau^*) y_p K(x, x_p) + \sum_{x_n \in SV^-, y_n < 0} \alpha_n y_n K(x, x_n) + b$$

Algorithm complexity

The runtime complexity of an SVM training using SMO method is $O(N^3)$, and evaluating is $O(N^2)$[5]. For the search strategy, runtime complexity is $O(log(1/\varepsilon))$[16]. For each τ, the gmean evaluation complexity is $O(N^2)$, so the runtime complexity of search τ is $O\left(\log\left(\frac{1}{\varepsilon}\right) * N^2\right)$. Hence the total runtime complexity of proposed method is $O(N^3)$.

IV. APPLICATION IN LINK PREDICTION IN CO-AUTHORSHIP NETWORKS

Co-authorship network or academic social network is a typical social network that can be formularized by a graph in which a node is an author or researcher and an edge reflects the connection between them in terms of having joint paper(s). Co-authorship network contains abundant academic characters in comparison with other social networks, so that analysing and mining information from co-authorship network have significant and practical meanings in modeling and increasing research quality [13]. Link prediction in co-authorship network is one of the important problems in social network research. Researchers have focused on analyzing and proposing solutions to give efficient recommendation to authors who can work together in a science project (e.g. a paper). Link prediction in the co-authorship network strengthens collaboration and idea exchange between scientists.

The aim of link prediction is to determine couples of authors who can collaborate in the future based on current network structure or on the information of authors, publish papers, etc. The model of link prediction problem can be described as follows:

Given a co-authorship network denoted by G^T=(V^T,E^T,P^T,T), where T=$\{t_1,t_2,\dots,t_k\}$ is a set of time stamps; V^T=$\{v_1,v_2,\dots,v_N\}$ is a set of nodes in T, each node represents an author in research community; P^T=$\{p_1,p_2,\dots,p_M\}$ is a

set of papers in T; $E^T = \{(v_i, v_j, p_k, t_h)\}$ is a set of links in T, two authors (v_i, v_j) wrote the paper p_k together in the time stamp t_h. Besides, the set V^T could contain attributes of each node, corresponding to the information of authors as nationality, university, research topics, etc. These attributes are denoted by $A^T = \{a_1, a_2, \ldots, a_N\}$, where a_i is a vector which includes information of author / node v_i.

The similarity metrics between two authors can be calculated from the attributes within the sets E^T and A^T. The link prediction problem may be modeled as below:

For two time intervals T_1 and T_2, where $T_1 < T_2$, let $G^{T_1 \cup T_2}$ denotes the network consisting of all edges with a time-stamp within T_1 and T_2. We choose the network G^{T_1} to a predictor; the predictor then outputs a list of edges which are not present in G^{T_1} but are predicted to appear in the network G^{T_2}. We refer to G^{T_1} as base graph (GB) and G^{T_2} as prediction graph (GR).

There are different approaches to solve the link prediction problem, but most preceding studies focused on in the network based on traditional metrics and then predicting the appearance of new links based on the data generated by the scores. The common similarity measures based on E^T may be listed as follows: The common neighbor score (CN) between two nodes u and v is measured by the number of common neighbors. The similarity score Adamic-Adar (AA) between u and v takes both the common neighbors and the common neighbors' neighbors into account. It can be expressed in follows: two actors are more similar if their common neighbors have less neighbors besides these two. Both the number of common neighbors and the number of total neighbors of two nodes are considered by Jaccard Coefficient score (JC). The Preferential Attachment score (PA) considers the multiplication of neighborhood size of two nodes as feature value. The score ShortestPath is the inverse of the shortest distance between two nodes. It means that collaboration is more likely if two nodes are close to each other. If there is no path between two nodes then this score has the value 0. The score Katz sums over all the paths that exist between a pair of nodes. However, the contribution of longer paths decreases by using an exponential factor. The equations for calculating similarity scores can be seen in [14]. Besides, there are other measures based on the set A^T which represented community scores of authors such as common nationality, common affiliate, common research topic, etc. Two authors from the same countries or from the same affiliates have a higher score of CommCountry, CommAffl. If they are interested in the same research topics then the score CommTopic is high.

Given a co-authorship network in the time intervals T_1 and T_2, where $T_1 < T_2$, then we can establish a dataset of candidates for two nodes for link prediction problem as follows:

	Similarity metrics in the time interval T_1	Linked label =1 (or = -1), if they are co-authors (or not co-authors) in the time interval T_2
The candidates as couple of nodes in time interval T_1	Values of similarity scores	Values of labels

Based on the above dataset, the link prediction in the co-authorship network may be considered as a binary classification problem which means that a pair of nodes (authors) can be classified into positive class (+1) or negative class (-1). If these nodes belong to positive class, they are able to have a new link in the future (in prediction). The similarity scores are the attributes as distinguished features of each candidate for classification. Hence, we can use a classification method like SVM to apply to the dataset. In the co-authorship network, the candidates of label -1 outnumber the candidates of other label, so it is an imbalanced dataset.

V. EXPERIMENTS

The experiments proposed method with three other techniques: standard SVM, Weighted-SVM, and z-SVM. As performance measures, we have used F1-score and G-mean measures to evaluate all methods. Parameter settings for each method:

- SVM: $C \in [2^{-7}, \ldots, 2^7]$, kernel: linear

- Weighted-SVM (WSVM): $\frac{C^-}{C^+} = \frac{N^+}{N^-}$, kernel: linear, N^+, N^- are numbers of patterns in positive and negative classes respectively.

- z-SVM: the initial value of z is $z^0 = 1$, kernel: linear, the optimal value z^* is solved by Golden section search

- Proposed method: $\frac{C^-}{C^+} = \frac{N^+}{N^-}$, the initial value of τ is $\tau^0 = 0$, $a = m = -||\alpha^+_{\max}||, b = M = ||\alpha^+_{\max}||, k = 10^{-3}, \varepsilon = 10^{-3}$ kernel: linear.

All algorithms were implemented in Matlab R2016a running environment.

A.Experiments on some imbalanced datasets from UCI[12]

In some first experiments, we tested on some binary-class imbalanced datasets from well-known UCI datasets, as presented in the table below:

Dataset	Dimension	Total	Positive(%Pos)	Negative(%Neg)	Train(Pos)	Test (Pos)
Abalone19	7	4174	32 (0.77%)	4143(99.23%)	3000(24)	1174(8)
Abalone9-18	7	731	42(6%)	689(94%)	500 (30)	231 (12)
Yeast	8	1484	304(20,5%)	1180(79.5%)	1000 (185)	484 (119)

Our selected datasets have different imbalanced data rate, from high to low. For each dataset, we split origin dataset into training and testing dataset but keeping the same imbalanced data rate in training and testing datasets. Training sets are used to train classification models and determine optimal value in z-SVM and the proposed method. Testing sets are used to evaluate the classification models. All results can be explained in the table below:

Datasets	Measures	SVM	z-SVM	WSVM	Proposed
Abalone19	F1-score	-	0.018	0.0453	**0.0585**
	G-mean	-	0.6439	0.7882	**0.8074**
Abalone9-18	F1-score	0.1538	0.5	0.5455	**0.6486**
	G-mean	0.2887	0.8358	0.9532	**0.9699**
Yeast	F1-score	0.5567	0.68	0.6875	**0.6904**
	G-mean	0.654	0.8195	**0.82**	0.8185

The above results of the experiments show that our proposed method improves the prediction better than that of other methods in terms of F1-score and G-mean, and has a good classification performance on imbalanced data.

B.Experiments on co-authorship networks

In these experiments, we built some candidate datasets from co-authorship networks data. Each candidate is calculated to obtain 7 similarity metrics, and assign label to indicate that candidate has collaboration actually work or not. Three datasets have differentthe the imbalanced data rates. The detail of datasets is presented in the table below:

Dataset	Dimension	Total	Positive(%Pos)	Negative(%Neg)	Train(Pos)	Test (Pos)
Co-authorship 1	7	5206	52(1%)	5154(99%)	3500 (35)	1706 (17)
Co-authorship 2	7	10417	947(9%)	9470(91%)	7700(700)	2717(247)
Co-authorship 3	7	2309	500(21,7%)	1809(78.3%)	1600 (340)	709 (160)

The scenario of training and testing in each method, also determine optimal value in z-SVM and proposed method are as same as the experiments in section A. In co-authorship networks, along with F1-score and G-mean measures, we also want to see the detail of TP, FP, TN, FN to observe the prediction of each method for each class label. All results can be represented as the table below:

Datasets	Measures	SVM	z-SVM	WSVM	Proposed
Co-authorship 1	TP	0	7	14	15
	FP	0	256	434	403
	TN	1654	1398	1220	1251
	FN	17	10	3	2
	F1-score	-	0.0500	0.0602	**0.0690**
	G-mean	-	0.5899	0.7794	**0.8169**
Co-authorship 2	TP	0	202	210	219
	FP	0	1093	728	728
	TN	2470	1377	1742	1742
	FN	247	45	37	28
	F1-score	-	0.2620	0.3544	**0.3668**
	G-mean	-	0.6752	0.7743	**0.7908**
Co-authorship 3	TP	40	115	128	136
	FP	21	137	167	173
	TN	528	412	382	376
	FN	120	45	32	24
	F1-score	0.362	0.5583	0.5626	**0.58**
	G-mean	0.4903	0.7344	0.7461	**0.763**

The above table lists the evaluation measures. From these results, it is possible to derive some conclusions:

- Standard SVM performs the worst. The number of false negatives (FN) in this method is the highest, which leads to low Recall, then also leads to low G-mean. For highly imbalanced dataset, the prediction of this method is poor.

- z-SVM performs better than standard SVM. However, the number of predicted positive samples is low. The F1-score and G-mean are also lower than that of Weighted-SVM.

- Weighted-SVM performs so good for imbalanced dataset. This method predicted better than standard SVM and z-SVM.

- The Proposed method outperforms the others in terms of TP, F1-score and G-mean measures. Specifically, by rectifying the skewness of the original classification hyperplane towards the positive samples, the number of false negatives (FN) in our proposed method is evidently reduced, thus enhancing the G-mean. In addition, the proposed method performs well when the training sample set is highly imbalanced.This indicates that our proposed method is suitable for prediction in co-authorship networks.

VI. CONCLUSION

This paper presents an improved SVM method through adjusting weighted vector w, while combining with the Weighted-SVM training method, to increase the efficiency of classification for imbalanced data and appling to link prediction problem in co-authorship networks. The weight vector is solved until a maximum G-mean measure value is achieved. Experimental results on various well-known datasets from UCI and co-authorship networks data with different ratios of imbalance verify that proposed method outperforms other SVM-based techniques when the training sample set is highly imbalanced.

VII. REFERENCES

[1] Corina Cortes, Vladimir Vapnik, Support-vector networks. Machine Learning, 20(3), 1995, pp. 273–297.

[2] Osuna, R. Freund, F. Girosi. Support vector machines: Training and applications. AI Memo 1602, Massachusetts Institute of Technology, 1997

[3] T. Imam, K.M. Ting, J. Kamruzzaman, "z-SVM: An SVM for Improved Classification of Imbalanced Data," Proc. Australian Joint Conf. Artif. Intell, Hobart, Australia, Dec. 4–8, 2006, pp. 264–273.

[4] Haydemar Núñez, Luis Gonzalez-Abril, Cecilio Angulo. Improving SVM Classification on Imbalanced Datasets by Introducing a New Bias, Journal of Classification, October 2017, Volume 34, Issue 3, pp 427–443

[5] John C. Platt (1998), Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines (PDF), CiteSeerX 10.1.1.43.4376

[6] http://www.neural-forecasting.com/support_vector_machines.htm

[7] https://en.wikipedia.org/wiki/Confusion_matrix

[8] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, Volume 2 Issue 3, April 2011.

[9] Rukshan Batuwita, Vasile Palade. Class Imbalance Learning Methods for Support Vector Machines, Imbalanced Learning: Foundations, Algorithms, and Applications, DOI: 10.1002/9781118646106.ch5, 10 June 2013

[10] G. Lee, H. Gurm, Z. Syed, Predicting Complications of Percutaneous Coronary Intervention using a Novel Support Vector Method. Journal of American Medical Informatics Association (JAMIA), 20(4):778-786, 2013

[11] Gill P. E., Murray W., Wright M. H.: Practical Optimization. Academic Press (1981)

[12] https://archive.ics.uci.edu/ml/

[13] Pham Minh Chuan, Le Hoang Son, Mumtaz Ali, Tran Dinh Khang, Le Thanh Huong, Nilanjan Dey, Link Prediction in Co-authorship Networks based on Hybrid Content Similarity Metric. Applied Intelligence, 48(8), 2018, pp. 2470-2486, ISSN: 0924-669X. Doi: 10.1007/s10489-017-1086-x.

[14] Phạm Minh Chuẩn, Trịnh Khắc Linh, Trần Đình Khang, Lê Hoàng Sơn (2017). Phân tích sự ảnh hưởng của một số độ đo liên kết áp dụng vào bài toán dự đoán liên kết trong mạng đồng tác giả. Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR) – Đà Nẵng, 17-18/8/2017. ISBN: 978-604-913-614-6, trang 760-767.

[15] R. Akbani, S. Kwek, N. Japkowicz, Applying Support Vector Machines to Imbalanced Datasets BT - Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 39–50.

[16] David G. Luenberger, Yinyu Ye, Linear and nonlinear programming. P.200. Google Books, 1984

PHƯƠNG PHÁP SVM CẢI TIẾN CHO DỮ LIỆU MẤT CÂN BẰNG VÀ ỨNG DỤNG CHO DỰ ĐOÁN LIÊN KẾT ĐỒNG TÁC GIẢ

Trịnh Khắc Linh, Trần Đình Khang, Phạm Minh Chuẩn

TÓM TẮT: *Trong bài toán phân lớp dữ liệu, sự mất cân bằng về lớp ảnh hưởng rất lớn đến hiệu quả của mô hình phân lớp. Đã có những nghiên cứu cải tiến SVM thích nghi với tập dữ liệu huấn luyện mất cân bằng. Bài báo này đề xuất một phương pháp phân lớp SVM cải tiến cho dữ liệu mất cân bằng bằng cách điều chỉnh vector trọng số w, đồng thời kết hợp với phương pháp huấn luyện weighted-SVM để tăng hiệu quả phân lớp cho dữ liệu mất cân bằng và áp dụng cho bài toán dự đoán liên kết đồng tác giả.*