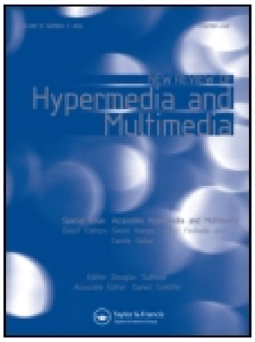

Taylor & Francis
Taylor & Francis Group

New Review of Hypermedia and Multimedia

ISSN: 1361-4568 (Print) 1740-7842 (Online) Journal homepage: http://www.tandfonline.com/loi/tham20

Label matrix normalization for semisupervised learning from imbalanced Data

Fengqi Li, Guangming Li, Nanhai Yang, Feng Xia & Chuang Yu

To cite this article: Fengqi Li, Guangming Li, Nanhai Yang, Feng Xia & Chuang Yu (2014) Label matrix normalization for semisupervised learning from imbalanced Data, New Review of Hypermedia and Multimedia, 20:1, 5-23, DOI: 10.1080/13614568.2013.846416

To link to this article: https://doi.org/10.1080/13614568.2013.846416

Published online: 06 Nov 2013.

New Review of Hypermedia and Multimedia, 2014
Vol. 20, No. 1, 5–23, http://dx.doi.org/10.1080/13614568.2013.846416

Label matrix normalization for semisupervised learning from imbalanced Data

FENGQI LI, GUANGMING LI, NANHAI YANG, FENG XIA* and CHUANG YU
School of Software, Dalian University of Technology, Dalian, China

(*Received 15 January 2013; final version received 16 September 2013*)

Manually labeled data-sets are vital to graph-based semisupervised learning. However, in the real world, labeled data-sets are often heavily imbalanced, and the classifiers trained on such skewed data tend to show poor performance for low-frequency classes. In this paper, we deal with an imbalanced data case of semisupervised learning and propose a novel label matrix normalization solution called LMN to tackle the general imbalance problem. Experiments over different data-sets reveal the effectiveness of the devised algorithm.

Keywords: Graph-based semisupervised learning; Imbalanced data; Label matrix normalization

1. Introduction

In contemporary society, the majority of information surges in short bursts and, as such, unlabeled data are abundant and can be easily obtained. Labeled data, though, are often very limited, and labeling a large number of data points requires tremendous human involvement and is very time consuming. Thus, it is necessary and promising to exploit methods that leverage both labeled and unlabeled instances. This motivates the hot research field of semisupervised learning (Zhu and Goldberg 2009, Zhang and Yeung 2012).

Semisupervised learning is a learning paradigm between supervised learning (Lou *et al*. 2012) and unsupervised learning (Angadi and Venkatesulu 2012). In this paper, we focus on semisupervised classification (SSC) and more specifically on graph-based SSC (GSSC) due to its reasonable model, intuitive expression, and strong comprehensibility. In GSSC, data samples (including labeled and unlabeled ones) can be viewed as nodes of a weighted graph, and the weighted edges reflect the similarity between the samples. Unlabeled data can get their labels via label propagation (Budvytis *et al*. 2010, Breve *et al*. 2012). Various GSSC methods (Zhu *et al*. 2003, Zhou *et al*. 2004, Jain *et al*. 2011, Qi *et al*. 2012) have been proposed in recent literatures, most of which are constructed under

*Corresponding author. Email: f.xia@ieee.org clustering and manifold assumptions (Zhu *et al*. 2003, Zhou and Li 2010, Urner *et al*. 2011). These assumptions are important in semisupervised learning, as the nearby data points or the data points forming the same manifold are likely to have the same label, or in other words, the labels are smooth on the graph. Based on data graph, Zhu *et al*. (2003) proposed an algorithm called Semisupervised Learning Using Gaussian Fields and Harmonic Functions (GFHF). In GFHF, a *k*-nearest-neighbor (*k*-NN) graph is constructed to carry out the label propagation. Zhou *et al*. 2004 utilized local and global consistency (LGC) to predict the labels of unlabeled samples on a completely connected graph.

In semisupervised learning, there usually comes this situation, different classes have different number of labeled data. For example, in a data-set of roses, some flowers are more common (red roses and white roses), while other species (green roses) are relatively scarce. Consequently, red and white roses may have more labeled data than the green ones, though all of the three categories may have the same amount of samples. This is usually called an “imbalanced circumstance” in the labeled data-set, and it will probably decrease the classification accuracy of semisupervised classifiers. However, traditional semisupervised learning algorithms assume a balance in a given labeled data-set, disregarding this realistic imbalance.

In this paper, a novel label matrix normalization method for semisupervised learning from imbalanced data, namely, LMN, is proposed. A normalization process is applied to the original label matrix to overcome the imbalance problem via balancing the total amount of information carried by labeled data in different classes. Some theoretical analysis is done, thereby proving our algorithm is promising for semisupervised applications. To summarize, the main contributions of this paper include the following:

(1) We provide a solution to the imbalance problem, which to some degree ensures classification accuracy.
(2) We propose a novel algorithm inspired by an existing algorithm, which can behave well even under different imbalance degrees.
(3) Extensive comparative experiments across different data-sets illustrate the effectiveness of our algorithm.

The rest of this paper is organized as follows. In Section 2, existing research and related work are expounded upon. In Section 3, our LMN technique is introduced and relevant theoretical analysis is outlined. Experiments are conducted in Section 4 to validate the effectiveness of LMN. Discussions are thoroughly presented in Section 5. Finally, Section 6 concludes the paper.

2. Related work

Some methods (Lee *et al*. 2011, Ghazikhani *et al*. 2012) have been proposed to solve the imbalance problem, which can be generalized into two categories: techniques at the data level and solutions at the algorithmic level. For clarity, the class with more labeled data are referred to as the mighty class (MC), and the class with fewer labeled data are referred as the weak class (WC).

2.1. Techniques at data level

In this section, we review the processing techniques at the data level. In such methods, resampling (Kuwadekar and Neville 2010) is the representative algorithm. It can be further divided into two branches, oversampling (Ghazikhani *et al*. 2012) and undersampling (Li *et al*. 2011). The undersampling algorithm randomly selects a subset of labeled data from the given labeled data of MC and then combines with the labeled data-set of WC to reform a balanced labeled data-set. Usually a cotraining (Zhang and Zhou 2011) method will follow to make use of the unlabeled data. At this point, some labeled data, which is usually useful in semisupervised learning will be "thrown out." In order to solve this problem, Li *et al*. (2011) proposed an improved method that performs the undersampling algorithm several times to obtain several subsets, and each subset will be reformed to a balanced labeled data-set with all WC sets. Accordingly, any existing semisupervised learning method (cotraining as usual) can be used to obtain several classifiers, which work together as an ensemble classifier. A drawback of this approach is that the iterative trainings and calculations will bring more complexity. On the contrary, oversampling methods reuse the labeled data of WC, combined with labeled data-sets of MC to reform a balanced labeled data-set. Kokiopoulou and Frossard (2010) proposed an idea of a "virtual sample," which provides another way of increasing labeled data in WC.

In addition, Chen and Mani (2011) used the active learning method to solve the imbalance problem by first dividing the data into two groups: labeled data and "to-label" data. They selected the most appropriate to-label data constantly to label first. Thus, they could increase the amount of labeled data of WC. But once the to-label data was labeled by mistake, including wrong data or a wrong label, the error information would accumulate and may affect the classifier's performance.

2.2. Solutions at algorithmic level

On the other hand, the solutions at the algorithmic level can be generalized as follows: (1) One-class classification (Lipka *et al*. 2012) tries to distinguish one class of objects from all other possible objects by learning from a training set containing only the objects of that class. With this method, one can complete the classification task of multiple classes one by one, and the labeled data-set does not have to be balanced. (2) Ozertem and Erdogmus (2011) proposed an algorithm called kernel self-consistent labeling. They used support vectors to transform the imbalance problem into a second coding question and then used the corresponding methods to solve the problem. (3) The bipartite ranking idea (Feng *et al*. 2012) and the method of multiview semisupervised learning (Li *et al*. 2012) also have been used to solve the imbalance problem. (4) Wang *et al*. (2008) proposed an algorithm called Graph Transduction via Alternating Minimization (GTAM) to improve the performance of classifiers by considering the relationship among different classes and updating the label matrix iteratively. However, further calculation is required to yield conclusively accurate results. In addition, once data are labeled falsely, the error information will spread throughout, thereby increasing the instability of the algorithm.

In our previous work (Li *et al.* 2013), we proposed a simple and effective approach to alleviate the unfavorable influence of imbalance problem by iteratively selecting a few unlabeled samples and adding them into the minority classes to form a balanced labeled data-set for the learning methods afterward. This paper is based on the observation that in GTAM a normalized label matrix Z is used to handle the imbalanced problem successfully. While considering the relationship among data samples, we propose here the LMN algorithm by simplifying the normalization. It neither changes the quantity of labeled data, nor introduces an additional training process. The key to our algorithm is to project the amount of label information in different classes into a same level.

3. LMN

Given the class label set C = {1, 2…, c}, where c represents the total number of class labels, and the data-set $X = \{x_1 \ldots x_l, x_{l+1} \ldots x_{l+u}\}$, where l represents the number of labeled data, having x_i ($i \in [1, l]$) has been labeled by y_i, where $y_i \in$ C, and u represents the number of unlabeled data (usually has $l << u$, $l + u = n$). The initial label matrix $Y = [Y_1^T, Y_2^T \ldots Y_n^T]^T \in R^{n\times c}$, means that if x_i has been labeled by j, then $y_{ij} = 1$, otherwise $y_{ij} = 0$.

The goal of semisupervised learning is to predict the labels of unlabeled samples. To reach this goal, we will use both labeled and unlabeled data.

3.1. Proposed algorithm

The imbalance appears in the initial label matrix **Y**, where the classes have different numbers of labeled samples. Due to the fact that each label has the same amount of information, then different classes have different total amount of label information, which in turn will affect the final classification accuracy.

To make the label information of each class reach equilibrium, we introduce a normalized matrix $U \in R^{c\times \mathrm{c}}$. U is a diagonal matrix, and the diagonal elements are defined as follows:

$$u_{\mathrm{ii}} = \frac{1}{Y_{\cdot i}^T \overrightarrow{1}} \tag{1}$$

where $Y_{\cdot i}^T$ represents the transpose of the i-th column of Y, and $\overrightarrow{1} = [1 \ldots 1]^T$. Thus, we define

$$Y' = YU \tag{2}$$

as the processed label matrix, and then we have $\sum_i Y'_{ij} = 1$, which means each class has the same level of label information at the beginning. In the following semisupervised learning, we use Y' to substitute Y for the label propagation and label distribution.

3.2. Rationality analysis

The core idea of LMN is to reduce the amount of information carried by labeled data of MC. Furthermore, we can achieve the goal of leveraging the label information in different classes to the same level of 1.

Assume that the labeled data x_i has the label c_i ($c_i \in [1, c]$), and x_j has the label c_j ($c_j \in [1, c]$), while i ≠ j. For any data $x_q \in X$, st. $q \neq i$, $q \neq j$, its label information in different classes at time t+1 can be expressed as follows:

$$I\left(x_{qi}\right)^{(t+1)} = \tau I\left(x_{qi}\right)^{(t)} + (1-\tau)\sum_{i=1}^{mi} I(x_i) w_{iq} \tag{3}$$

$$I\left(x_{qj}\right)^{(t+1)} = \tau I\left(x_{qj}\right)^{(t)} + (1-\tau)\sum_{j=1}^{mj} I\left(x_j\right) w_{jq} \tag{4}$$

where $I(x_{qi})^{(t+1)}$ represents the label information of class c_i at time $t + 1$, and $I(x_{qj})^{(t+1)}$ has the similar meaning. $I(x_{qi})^{(t)}$ represents the label information of class c_i at time t, and $I(x_{qj})^{(t)}$ has the similar meaning. $I(x_i)$ represents the label information of class c_i carried by x_i and $I(x_j)$ has the similar meaning. m_i represents the total number of labeled data with label c_i, and these labeled data must be in relationship with x_q. m_j has the similar meaning. The parameter τ is used to balance these two terms. The data x_q will be labeled with class label c_q, only if label c_q has the largest amount of information than any other labels as far as data x_q.

In GTAM, a matrix Z is introduced to normalize the label matrix Y, and inspired by this, we simplify its processing technique by Equation (1). Then the label information of class c_i accumulated by data x_q is not concerned with x_j. Generally speaking, the label information of class c_i for a certain data can change independently despite the influence of class c_j, while $i \neq j$. Thus, for labeled data, only information of a determined class is available at time 0, and no label information can be transferred from other data to them. For already labeled data, label change may occur and depend only on its determined class. Furthermore, only internal change in each class is reasonable. So we can change the amount of label information within a certain class without considering the influence from other classes. In other words, Equation (1) is reasonable.

The goal of unifying the label information into one is to prevent additional complexity in the algorithm. Supposing that the amount of label information is unified to the size of the smallest labeled data-set, not one, it is necessary to find a certain class with least labeled data. This will result in additional complexity and overhead in the algorithm.

3.3. Semisupervised learning

As mentioned in the introduction, the weighted edges on the graph reflect the similarity between data samples. The similarity matrix (also called the weight matrix) $W \in R^{n\times n}$ can be calculated by the following classic formula:

$$w_{ij} = \exp\left(-||x_i - x_j||^2/\sigma^2\right) \tag{5}$$

If x_i is one of x_j's nearest k neighbors, or x_j is one of x_i's nearest k neighbors, the similarity between x_i and x_j ought to be calculated by Equation (5) and zero otherwise. The $||\cdot||$ in Equation (5) means the 2-norm of vector. σ is a relevant parameter, and for different data-sets, it has different values. Usually, we consider the maximum (or a bit larger than the maximum) in the characteristics matrix to be the value of σ for higher accuracy.

Before using the weight matrix, similar to what Nie *et al.* (2010) did, we first do some normalization of W to ensure that the final output be probabilities of the data points belonging to the classes (Zhou and Scholkopf 2004). More specifically, we calculate a normalized weight by

$$\bar{w}_{ij} = \frac{w_{ij}}{\sqrt{d_i d_j}}, \tag{6}$$

where

$$d_i = \sum\nolimits_j w_{ij}.$$

For the convenience of further calculation, we normally project the weight matrix $\bar{W}$ into the following expression:

$$\bar{W} = \begin{bmatrix} \bar{W}_u & \bar{W}_{lu} \\ \bar{W}_{ul} & \bar{W}_{uu} \end{bmatrix},$$

where the labeled l data are in the upper-left corner of the matrix and the unlabeled u data are in the lower-right corner. Then we will get the propagation matrix by

$$P = \bar{D}^{-1}\bar{W}, \tag{7}$$

where $\overline{D}$ is a diagonal matrix with entries

$$\bar{d}_i = \sum\nolimits_j \bar{w}_{ij}.$$

As so far, we can get the final classifier F in an iterative way:

$$F(t{+}1) = \alpha PF(t) + (1-\alpha)Y' \tag{8}$$

where α is a parameter in (0,1) and Y' is the normalized label matrix. By the iteration in Equation (8), we have

$$F(t) = (\alpha P)^{t-1}Y' + (1-\alpha)\sum_{i=0}^{t-1}(\alpha P)^i Y'.$$

Hence,

$$\lim_{t\to\infty}(\alpha P)^{t-1} = 0 \ \lim_{t\to\infty}\sum\nolimits_{i=0}^{t-1}(\alpha P)^i = (I-\alpha P)^{-1}$$

and then

$$\lim_{t\to\infty} F(t) = I_\beta(I-\alpha P)^{-1}Y', \tag{9}$$

which is equivalent to $(I{-}\alpha P)^{-1}Y'$. I_β equals to $(1{-}\alpha)I$.

As aforementioned, most graph-based semisupervised learning algorithms are based on the clustering and manifold assumptions. Then they essentially estimate a function (usually called the evaluation function) over the graph such that it satisfies two requirements: (1) labeled data can get correct labels and (2) labels on the graph are smooth. These two terms are often called the loss function and smoothness constraint. The more optimal the learning algorithm, the smaller the value of the evaluation function will be. In LMN, the evaluation function is defined as follows:

$$Q(F) = \frac{1}{2}\left(\sum\nolimits_{ij=1}^{n} \bar{W}_{ij}\left\|F_i - F_j\right\|^2 + \mu\sum\nolimits_{i=1}^{n} \bar{d}_i\left\|F_i - Y'_i\right\|^2\right) \tag{10}$$

The two terms in Equation (10) represent the smoothness constraint and loss function, respectively. In the second term, Y' is the normalized label matrix calculated by Equation (2). The parameter μ balances the importance between these two terms, and when it tends to be ∞, the function Q becomes to a harmonic function. Equation (10) can be rewritten as follows:

$$Q(F) = \mathrm{tr}\left(F^T LF\right) + \mathrm{tr}\left((F - Y')^T V\bar{D}(F - Y')\right) \tag{11}$$

where L represents the Laplacian operator and is defined as $L = \bar{D} - \bar{W}$. The notation *tr*()represents the trace of a matrix. The matrix V indicates a diagonal matrix with the parameter μ as its entries.

The optimal solution for the classification matrix F can be easily obtained by setting the derivative of *Q(F)* to zero. Therefore, we get

$$\frac{\partial Q}{\partial F} = 0 \Rightarrow LF^* + V\bar{D}(F^* - Y) = 0$$

and then F^* can be expressed as follows:

$$\begin{aligned} F^* &= (L + V\bar{D})^{-1} V\bar{D}Y' \\ &= (I - P + V)^{-1} VY' \\ &= (I - \alpha P)^{-1} I_\beta Y'. \end{aligned} \tag{12}$$

This is the same as the final result of Equation (8).

While our algorithm gets the optimal final label matrix as Equation (12), different expressions of the optimal classification matrix F^* can be deduced from different algorithms. The LGC algorithm (Zhou *et al.* 2004) uses an evaluation function similar to Equation (10) and its corresponding F^* can be expressed as follows:

$$S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$$F^* = (I - \alpha S)^{-1} Y'$$

where I is an $n \times n$ identity matrix.

Meanwhile, the GFHF algorithm (Zhu *et al.* 2003) does not have the loss function term in its evaluation function, and its corresponding optimal classification matrix $\boldsymbol{F^*}$ has the expression as follows:

$$F_u^* = (D_{uu} - W_{uu})^{-1} W_{ul} F_l^*$$

where F^* is decomposed into:

$$F^* = \begin{bmatrix} F_l^* \\ F_u^* \end{bmatrix}$$

One can change the form of the evaluation function to solve particular problems. For example, Sebe *et al.* (2011) rewrote the evaluation function into three terms to

Table 1. Workflow of our algorithm.

```
Algorithm 1: LMN for Semisupervised Learning.

Input
  Data-set X, label set C and original label matrix $\mathbf{Y}$
Algorithm
  1. Calculate $\mathbf{Y}'$ (1) and (2)
  2. Calculate weight matrix and its normalization by (5) $\sim$ (6)
  3. Calculate the propagation matrix $\mathbf{P}$ by (7)
  4. Construct a $k$-NN graph
  5. Get the final classifier by (9)
Output
  Labels for data samples in X
```

express the geometry of support vectors, and the additional term was introduced to control the complexity of the algorithm to avoid the overfitting phenomenon.

3.4. Algorithm workflow

Our algorithm can be divided into two main phases, a LMN phase and a semisupervised learning phase. Table 1 describes the workflow of our algorithm.

If the classifier gives the data x_i the label j, then y'_{ij} will be 1, otherwise, it will have a value of 0. Furthermore, we can use the final label matrix Y' to calculate the classification accuracy.

4. Experiments

To verify the effectiveness of our algorithm, the LMN method was evaluated over different data-sets. These included the popular TOY data-set, the University of California, Irvine (UCI) standard data-sets (Bache and Lichman 2013) IRIS, WINE, and United States Postal Service (USPS). To better reflect the practicality of our algorithm, we conduct experiments on GENE, MOVEMENT and VOWEL data-sets (practical data-sets), which include the information about gene, gesture, and vowel, respectively.

To verify the imbalance processing capability of LMN, we will complement three experiments on TOY data-set, three UCI standard data-sets, and three practical datasets, respectively. Moreover, we conduct another set of experiments to verify our algorithm's performance when it is applied to the balanced situation. The comparative experimental results explain whether the LMN algorithm is better than the imbalance processing technique in GTAM.

4.1. Data-sets overview

We will use seven data-sets in this section, and the brief descriptions of the specific circumstances of each data-set are just as follows:

- TOY data-set—A data-set constructed by sine and cosine functions; has two classes with 240 data items each. Due to its meniscus expression, it has become one of the most popular data-sets in machine learning.

- USPS data-set—A data-set describes the American postal handwritten numerals, which has 10 classes. Each class has a different number of instances consisting of 256 characteristics. To conduct our experiment, we chose 200 instances from each class to reform a data-set with 2000 instances.
- WINE data-set—A data-set about three kinds of grape wine, which has 59, 71, and 48 instances consisting of 13 characteristics. Notice that the data-set even has an imbalance problem, and to ensure there's only one variable, we chose 48 samples from each class.
- IRIS data-set—A data-set describes three categories of iris. Each class has 50 instances and consists of four characteristics. This data-set is given further consideration as the characteristics information of two categories is extremely similar, so the performance of LMN is scrutinized in this special case.
- GENE data-set—A data-set consists of 105 gene sequences. Each has 57 basic groups (A, C, G, and T). It has two classes, donator and acceptor, with 53 instances, respectively. Different classes have different number of basic groups, so we use the number of different basic groups to describe instances.
- MOVEMENT data-set—A data-set which describes gestures in 15 different classes. Each class has 24 instances and consists of 90 characteristics.
- VOWEL data-set—A data-set which uses 10 different characteristics to describe 11 vowels. Each vowel is read by 15 individuals six times, so it has 90 instances.

4.2. Experiment settings

In order to verify the capacity of our algorithm under different imbalance degrees, we need to distribute different classes into groups in some cases. The distribution settings are as follows. On USPS, number 0 and 1 construct a group, and number 2–5 will be together, as well as number 6–9 forms another one. On MOVEMENT, the gesture category 1–5 can construct a group, and gesture 6–10 forms another one, as well as the remaining gesture 11–15 will be combined. On VOWEL, vowel 0–3 can be a group, and vowel 4–7 will be combined, as well as vowel 8–10 from the last group. For data-sets with two classes, the imbalance degree changes from 1:10–10:10. And for data-sets with three class groups, the imbalance degree varies from 2:6:10, 4:7:10–10:10:10. We will examine the LMN algorithm under different imbalance degrees to evaluate its performance.

4.3. Results on TOY data-set

The TOY data-set can be expressed by two meniscus shapes as Figure 1 shows. The abscissa means the input variable for sine function and cosine function varies in [0, 3.7] and [1.5, 5], respectively. Following the experimental settings, we select 1–10 labeled data, respectively, from the class blue rectangle (the one above) and 10 labeled data from class red circle (the one blow) all the time. Thus we get different labeled data-sets with different imbalance degrees. The GTAM algorithm itself has the ability to solve the imbalance problem, so we use it as the baseline algorithm here for comparison. The corresponding experimental results are depicted in Figure 2. To examine whether the number of labeled data under

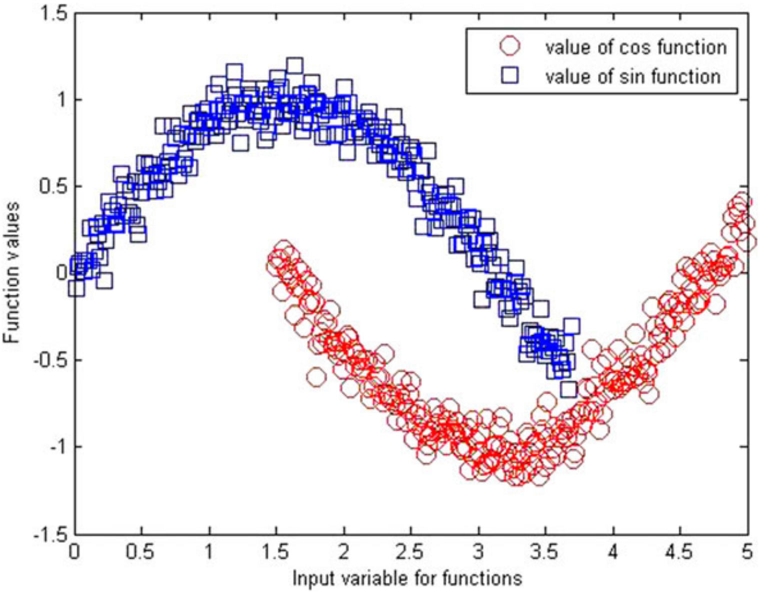

Figure 1. Meniscus shapes of TOY data-set.

the same imbalance degree will affect the performance of LMN, we conduct another set of experiments. Herein the labeled data from two classes change from 2:20, 4:20 to 20:20, with the same imbalance degree as from 1:10 to 10:10. The experimental results accordingly are shown in Figure 3.The abscissa represents the number (or half of the number in Figure 3) of labeled data from class blue rectangle.

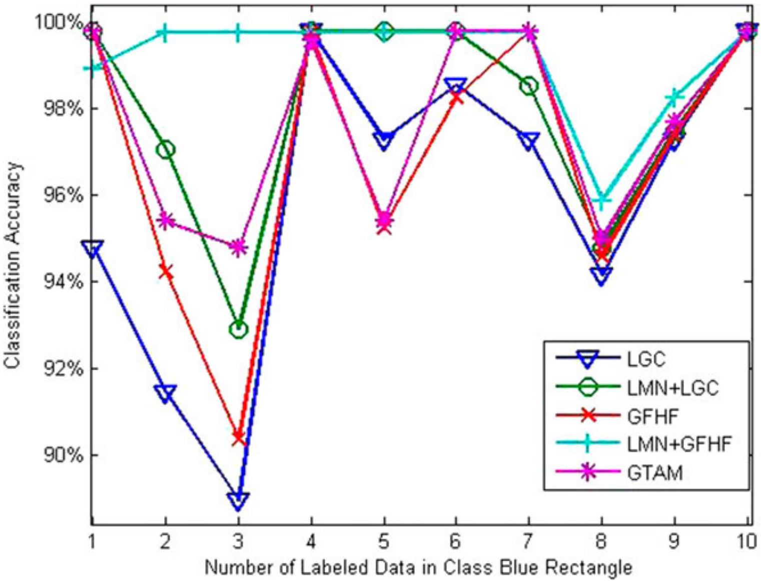

Figure 2. Accuracies on TOY data-set.

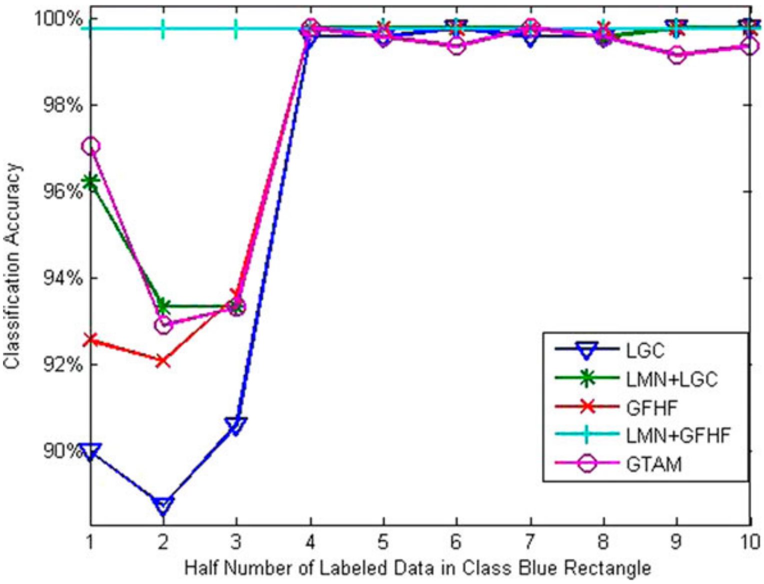

Figure 3. Accuracies under same imbalance degrees.

The experimental results show that the LGC algorithm performs worse than the other methods on an average, while GFHF enhanced by LMN has the best performance. The classification accuracy of LGC algorithm will be increased by 6% at most and 1.8% on an average when enhanced by LMN. Likely, GFHF will be increased by 8% at most and 2% on an average. Both the figures show that sometimes GTAM behaves better than almost any other algorithm, and sometimes, it will have the worst performance. This fact verifies our aforementioned analysis that GTAM is rendered instable upon the occurrence of false labels.

Then we can get conclusions such as: (1) LMN results in improved performance of the LGC and GFHF algorithms when imbalance problem occurs; (2) LMN has sound performance no matter what the imbalance degree will be; (3) Different numbers of labeled data under the same imbalance degree do not affect the performance of LMN. Therefore, in the following experiments, we show the results based on the previous experimental settings. Overall, LMN behaves well on TOY data-set.

4.4. Results on UCI data-sets

We evaluate the LMN algorithm as well as LGC and GFHF on UCI standard data-sets. The GTAM algorithm is also used as a baseline for comparison. Figures 4–6 display the classification accuracies on different data-sets. According to the previous experimental settings, when the abscissa tends to *i*, we select 2**i*, *i*+5 and 10 labeled data from different class groups, respectively.

As shown in Figures 4–6, it is inferred that LMN improves the classification accuracy of LGC and GFHF under different imbalance degrees. On IRIS data-set,

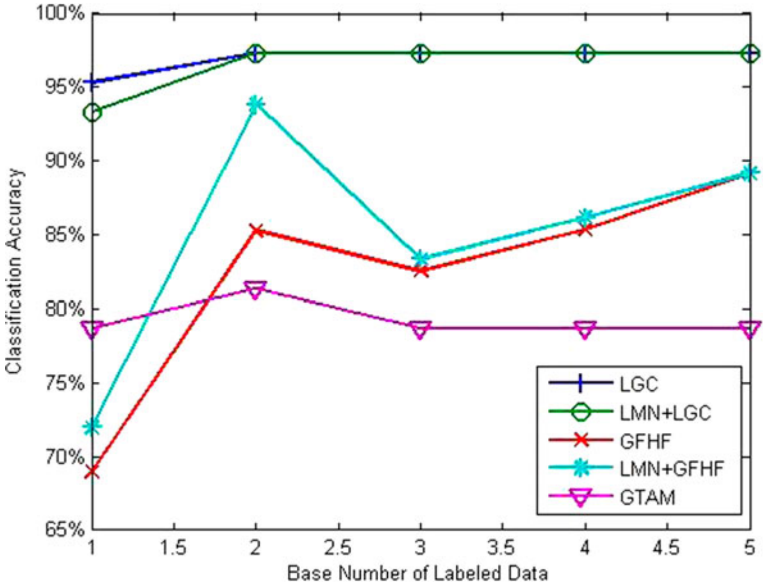

Figure 4. Classification accuracies on IRIS data-set.

classification accuracies of LGC and GFHF are increased by 2% and 8% at most, respectively. The percentages are 1%, 4% and 6%, 2% on WINE and USPS data-sets. Moreover, classification accuracies under the enhancement by LMN are over 20% higher than GTAM. Overall, LMN performs well on UCI standard data-sets.

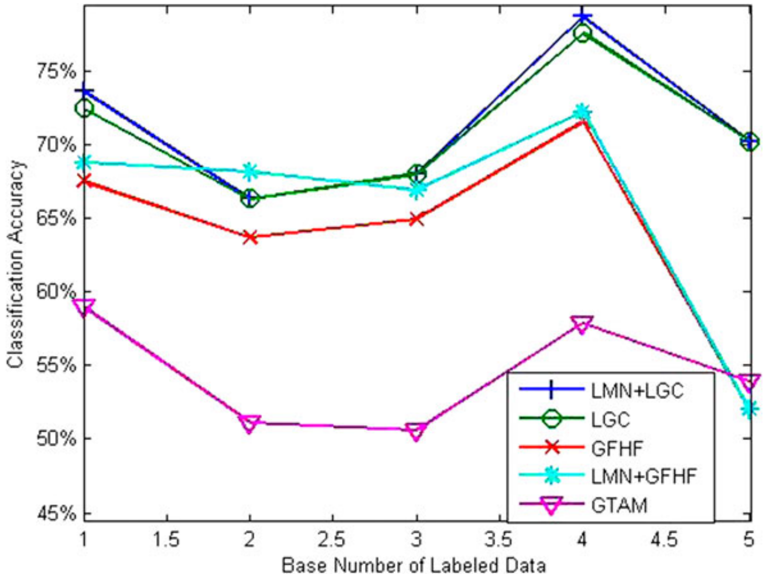

Figure 5. Classification accuracies on WINE data-set.

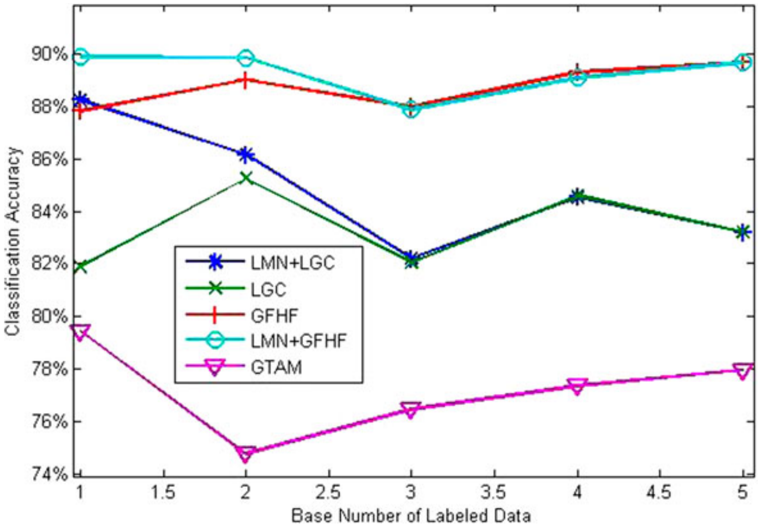

Figure 6. Classification accuracies on USPS data-set.

4.5. Results on practical data-sets

On practical data-sets with imbalance problem, LMN can also bring improvements to LGC and GFHF, as shown in Figures 7–9. The 11 classes in the VOWEL data-set have similar features. So the experimental results can also examine whether our algorithm can behave well when all classes are similar to each other.

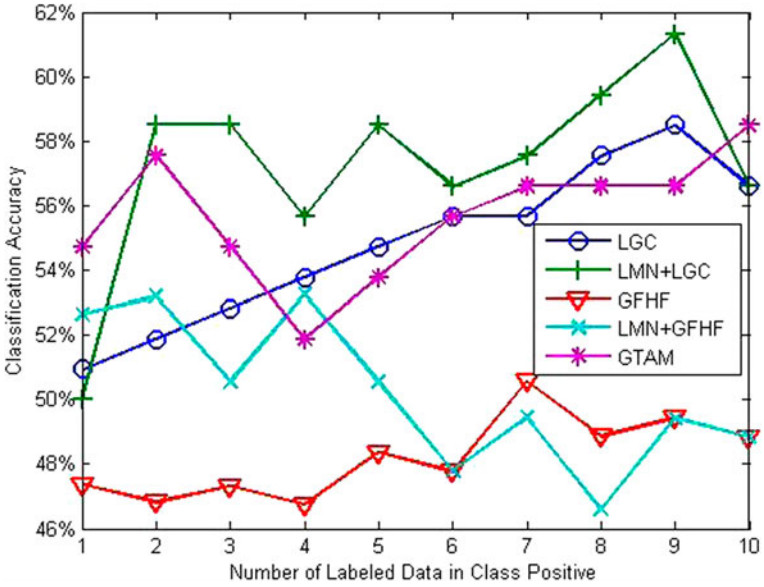

Figure 7. Classification accuracies on GENE data-set.

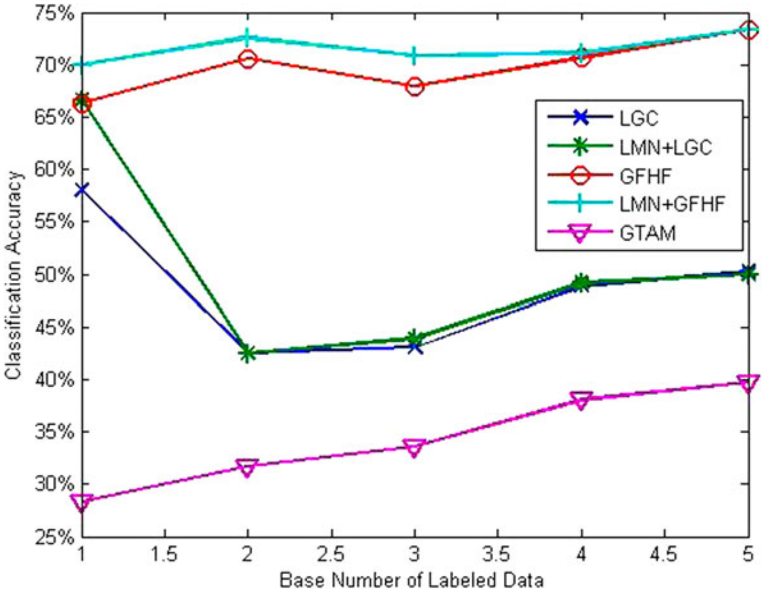

Figure 8. Classification accuracies on MOVEMENT data-set.

The experimental results show LMN behaves well most of the time. But on GENE data-set, there are two exceptions for LMN + GFHF as compared to GFHF. Almost all algorithms except LGC have instability. Even though, the classification accuracies of LGC and GFHF are increased by 2.5 and 2% as an

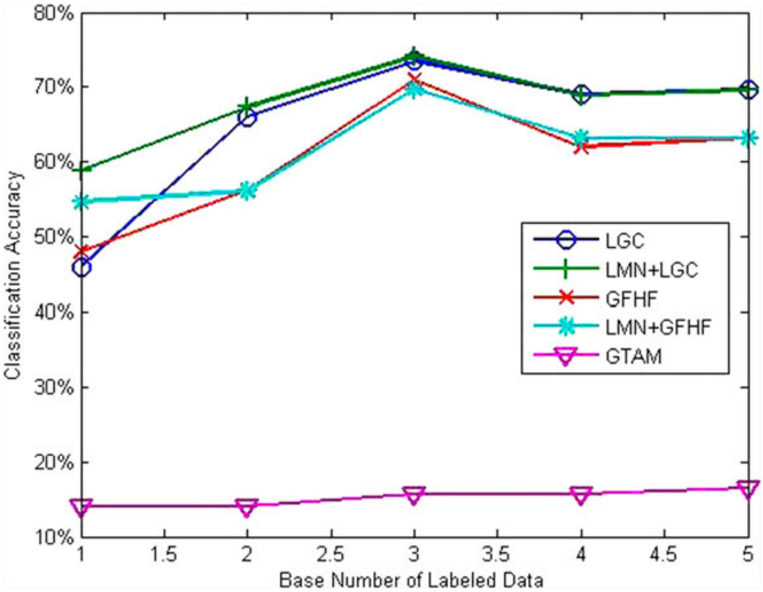

Figure 9. Classification accuracies on VOWEL data-set.

average when enhanced by LMN, respectively. On MOVEMENT and VOWEL data-sets, the experimental results do not reveal instability and the improved algorithms behave much better than GTAM. The increased percentages become 1.3%, 2.5% and 3%, 4%. To summarize, LMN performs well on practical data-sets.

4.6. Results in balanced situations

In addition to the imbalance problem addressed above, the performance of LMN in balanced situations should be examined. For this purpose, we conduct experiments using the IRIS data-set (other data-sets will yield similar results). Each class has 10 labeled data. Figures 10–12 show the results. It is observed that the performance of all the examined algorithms is comparative. The classification accuracies in percentage are very close in most cases.

4.7. Comparison against GTAM

In most figures above, we can see GTAM has worse performance than LGC and GFHF + LMN. As GTAM uses its own imbalance processing and learning techniques, we cannot judge whether LMN is superior to GTAM as an imbalance processing approach. To investigate this, we conduct another set of experiments on WINE data-set (any other data-sets used in this paper will lead to the same results), and the learning method of LGC will be used. Figure 13 shows the experimental results. It reveals that our algorithm performs better than GTAM. Note that GTAM + LGC means combining the imbalance processing approach in GTAM with the LGC learning method.

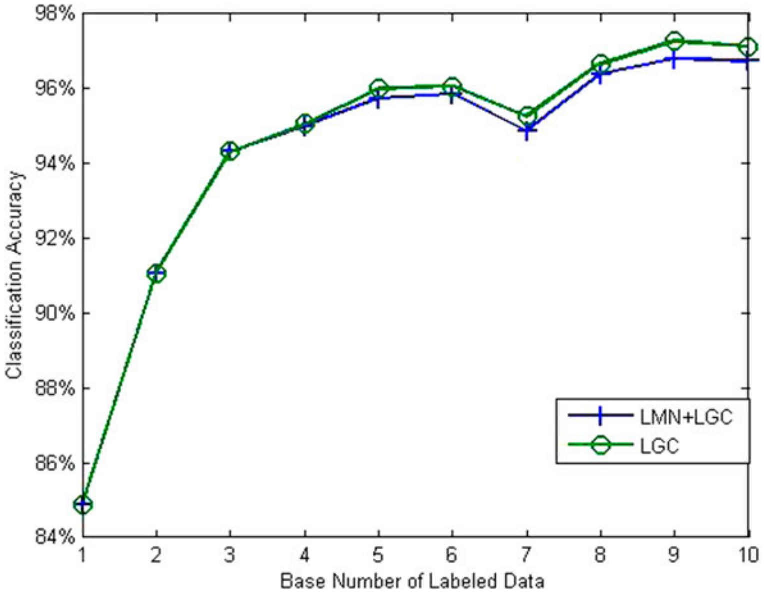

Figure 10. LMN + LGC vs LGC.

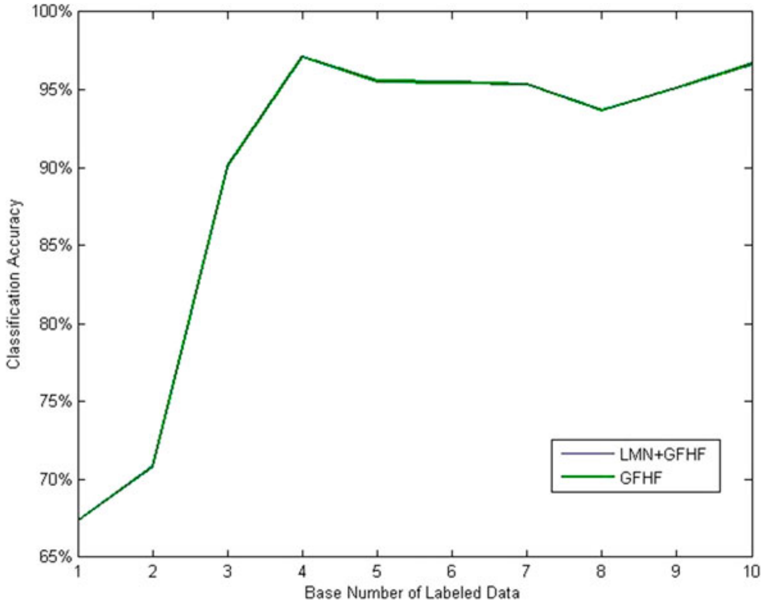

Figure 11. LMN + GFHF vs GFHF.

5. Discussions

When constructing imbalanced labeled data-sets, we consider the different imbalance degrees, and change it from 1:10 to 10:10. Due to the randomness of the labeled data and their class labels, LMN algorithm will have good

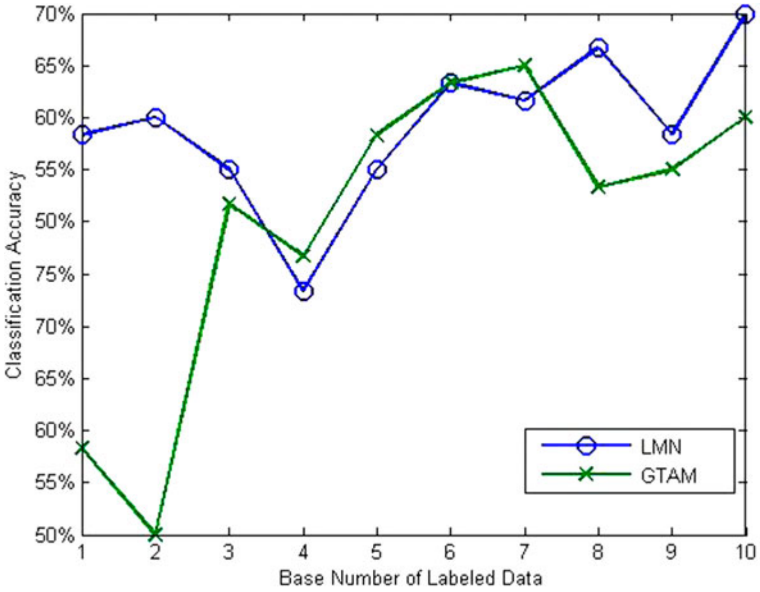

Figure 12. LMN vs GTAM.

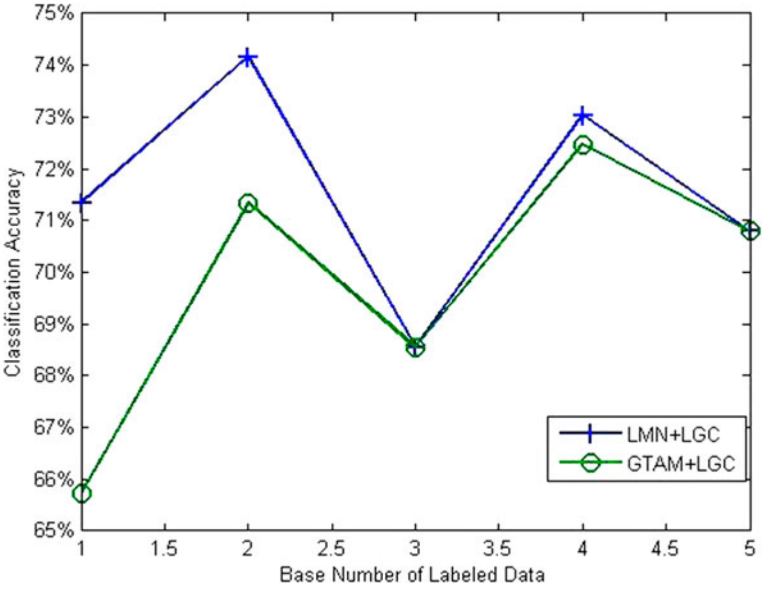

Figure 13. GTAM vs LMN.

performance when we choose another option, for example, changing the imbalance degree from 10:1 to 10:10.

It is easy to see that our algorithm simply changes the initial information of labeled data. Consequently, LMN will not increase the time complexity considerably. However, LMN also brings much instability in some cases.

While LMN projects the label information of different classes into the same level, it has its own defects. First, its performance is not good enough in the situation of balanced information. The label information of MC was strongly reduced at the beginning. Second, the LMN method may be easily influenced by noise data. Finally, the convergence speed of LMN may decreases because the reduction of label information may cause some nodes to be unable to determine their class labels.

Some parameters of algorithms, like the value of k in k-NN graph, and the σ in Equation (5), and so on, are eternal issues. We chose the optimal value in experiments empirically, such as k will get the value of 6. Different data-sets should have different values of σ. In the experiments performed, the maximum value in the characteristics matrix was selected to be the value of σ.

6. Conclusions

In this paper, we have proposed the LMN algorithm to solve the imbalance problem. It is inspired by a similar normalization in the GTAM algorithm. In particular, LMN projects the label information of different classes into a same level, and it will neither replicate erroneous information, nor bring additional complexity to the algorithm. Experimental results across different data-sets illustrate the effectiveness of our algorithm.

Acknowledgments

This work was partially supported by the Fundamental Research Funds for Central Universities (DUT12JR10) and Liaoning Provincial Natural Science Foundation of China under Grant No. 201202032.

References

U.B. Angadi and M. Venkatesulu, "Structural SCOP superfamily level classification using unsupervised machine learning", *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(2), pp. 601–608, 2012.

K. Bache and M. Lichman, *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science, 2013. Available online at: http://archive.ics.uci.edu/ml.

F. Breve, L. Zhao, M. Quiles, W. Pedrycz and J.M. Liu, "Particle competition and cooperation in networks for semi-supervised learning", *IEEE Transactions on Knowledge and Data Engineering*, 24(9), pp. 1686–1698, 2012.

I. Budvytis, V. Badrinarayanan and R. Cipolla, "Label propagation in complex video sequences using semi-supervised learning", *British Machine Vision Conference (BMVC)*, British Machine Vision Association, Manchester, England, pp. 2257–2263, 2010.

Y. Chen and S. Mani, "Active learning for unbalanced data in the challenge with multiple models and biasing", *Journal of Machine Learning Research*, 6, pp. 113–126, 2011.

J. Feng, X. He, B. Konte, C. Bohm and C. Plant, "Summarization-based mining bipartite graphs", in *Proceedings of the 18th ACM SIGKDD International Conference on KDD*, New York: ACM. pp. 1249–1257, 2012.

A. Ghazikhani, H.S. Yazdi and R. Monsefi, "Class imbalance handling using wrapper-based random oversampling", in *Proceedings of 20th Iranian Conference on Electrical Engineering*, Washington, DC: IEEE, pp. 611–616, 2012.

R. Jain, T.S. Chua, J.H. Tang, R. Hong, S.C. Yan and G.J. Qi, "Image annotation by kNN-sparse graph-based label propagation over noisily tagged web images", *ACM Transactions on Intelligent Systems and Technology*, 2 (2), pp. 1–15, 2011.

E. Kokiopoulou and P. Frossard, "Graph classification of multiple observation sets", *Pattern Recognition*, 43 (12), pp. 3988–3997, 2010.

A. Kuwadekar and J. Neville, "Combining semi-supervised learning and relational resampling for active learning in network domains", in *Proceedings of the Budget Learning Workshop, 27th International Conference on Machine Learning*, Haifa, Israel: ACM, 2010.

S.Y. Lee, S.S. Li, G.D. Zhou and R.Y. Wang, "Imbalanced sentiment classification", in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, New York: ACM, pp. 2469–2472, 2011.

G.X. Li, K.Y. Chang and C.H. Steven, "Multi-view semi-supervised learning with consensus", *IEEE Transactions on Knowledge and Data Engineering*, 24, pp. 2040–2051, 2012.

S.S. Li, Z.Q. Wang, G.D. Zhou and S.Y. Lee, "Semi-supervised learning for imbalanced sentiment classification", in *Proceedings of the Twenty-Second International Joint Conference on Artificial intelligence*, New York: ACM. pp. 1826–1831, 2011.

F.Q. Li, C. Yu, N.H. Yang, F. Xia, G.M. Li and F. Kaveh-Yazdy, "Iterative nearest neighborhood oversampling in semisupervised learning from imbalanced data", *The Scientific World Journal*, 2013, 9 p, 2013.

N. Lipka, B. Stein and M. Anderka, "Cluster-based one-class ensemble for classification problems in information retrieval", in *Proceedings of the 35th international ACM SIGIR conference on Research and Development in Information Retrieval*, New York: ACM, pp. 1041–1042, 2012.

Y. Lou, R. Caruana and J. Gehrke, "Intelligible models for classification and regression", in *Proceedings of the 18th ACM SIGKDD international conference on KDD*, New York: ACM, pp. 150–158, 2012.

F.N. Nie, S.M. Xiang, Y. Liu and C.S. Zhang, "A general graph-based semi-supervised learning with novel class discovery" *Neural Computing & Applications*, 19(4), pp. 549–555, 2010.

U. Ozertem and D. Erdogmus, "Locally defined principal curves and surfaces", *Journal of Machine Learning Research*, 12(4), pp. 1249–1286, 2011.

Z.Q. Qi, Y.J. Tian and Y. Shi, "Laplacian twin support vector machine for Semi-supervised classification", *Neural Networks*, 35, pp. 46–53, 2012.

N. Sebe, J. Uijlings, F.P. Nie, Y. Yang and Z.G. Ma, “Exploiting the entire feature space with sparsity for automatic image annotation”, in *Proceedings of the 19th ACM International Conference on Multimedia*, New York: ACM, pp. 283–292, 2011.

R. Urner, S.B. David and S.S. Shwartz, “Access to unlabeled data can speed up prediction time”, in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, Washington, DC: IEEE, pp. 641–648, 2011.

J. Wang, T. Jebara and S.F. Chang, “Graph transduction via alternating minimization”, In: *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, New York: ACM, pp. 1144–1151, 2008.

Y. Zhang and D.Y. Yeung, “Transfer metric learning with semi-supervised extension”, *ACM Transaction on Intelligent Systems and Technology (TIST)*, 3(3), pp. 1–28, 2012.

M.L. Zhang and Z.H. Zhou, “Co-trade: Confident co-training with data editing”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41, pp. 1612–1626, 2011.

D. Zhou, O. Bousquet, T.N. Lal, J. Weston and B. Schlkopf, “Learning with local and global consistency”, *Proceedings of Neural Information Processing Systems* (*NIPS)*, 16(16), pp. 321–328, 2004.

Z.H. Zhou and M. Li, “Semi-supervised learning by disagreement”, *Knowledge and Information Systems*, 24(3), pp. 415–439, 2010.

D. Zhou and B. Scholkopf, “Normalized cuts and image segmentation”, *IEEE Transactions on PAMI*, 22(8), pp. 888–905, 2004.

X.J. Zhu, Z. Ghahramani and J. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions”, in *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, Palo Alto, CA: Association for the Advancement of Artificial Intelligence (AAAI), pp. 912–919, 2003.

X.J. Zhu and A.B. Goldberg, “Introduction to semi-supervised learning”, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, San Rafael, CA: Morgan and Claypool, pp. 1–130, 2009.