Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(3): 2432–2450.
DOI: 10.3934/mbe.2020132
Received: 18 November 2019
Accepted: 15 January 2020
Published: 17 February 2020

*Research article*

# Label propagation algorithm based on Roll-back detection and credibility assessment

**Ying Dong, Wen Chen*, Hui Zhao*, Xinlei Ma, Tan Gao and Xudong Li**

College of Cyber Security, Sichuan University, Chengdu 610065, P.R. China

* **Correspondence:** Email: wenchen@scu.edu.cn; zhaohui@scu.edu.cn; Tel: +86-28-85405568.

**Abstract:** The traditional label propagation algorithm (LPA) iteratively propagates labels from a small number of labeled samples to many unlabeled ones based on the sample similarities. However, due to the randomness of label propagations, and LPA's weak ability to deal with uncertain points, the label error may be continuously expanded during the propagation process. In this paper, the algorithm label propagation based on roll-back detection and credibility assessment (LPRC) is proposed. A credit evaluation of the unlabeled samples is carried out before the selection of samples in each round of label propagation, which makes sure that the samples with more certainty can be labeled first. Furthermore, a roll-back detection mechanism is introduced in the iterative process to improve the label propagation accuracy. At last, our method is compared with 9 algorithms based on UCI datasets, and the results demonstrated that our method can achieve better classification performance, especially when the number of labeled samples is small. When the labeled samples only account for 1% of the total sample number of each synthetic dataset, the classification accuracy of LPRC improved by at least 26.31% in dataset circles, and more than 13.99%, 15.22% than most of the algorithms compared in dataset moons and varied, respectively. When the labeled samples account for 2% of the total sample number of each dataset in UCI datasets, the accuracy (take the average value of 50 experiments) of LPRC improved in an average value of 23.20% in dataset wine, 20.82% in dataset iris, 4.25% in dataset australian, and 6.75% in dataset breast. And the accuracy increases with the number of labeled samples.

**Keywords:** Label propagation; small number of labeled samples; credibility assessment; certainty; roll-back detection

## 1. Introduction

Traditional machine learning algorithms can be generally divided into three categories: (1) Supervised learning, which aims to learn from a large number of labeled samples and predicts the label of new unknown samples from the learned knowledge. In the practice, usually we can collect a large number of unlabeled samples, but the process of assigning labels to the samples is time consuming [1]. (2) Unsupervised learning, there is no corresponding category known in advance, so the classifier can only be learned from the sample set without labels. The unsupervised learning models are usually built based on the similarity between samples. (3) Semi-supervised learning, it relies on a small number of labeled samples to guide the label prediction of unlabeled samples [2,3]. New labeled samples are continuously added to the training set to compensate for the small number of labeled samples, which leads to performance defects in supervised learning.

Recently, the application of semi-supervised learning algorithm is more and more extensive, and scholars have done a lot of researches in this area. In addition to the traditional semi-supervised learning algorithm, such as S3VM [4], many new methods have been proposed. J. Levatic, et al. [5] proposed an extension of predictive clustering trees for multi-target regression (MTR) and ensembles thereof towards semi-supervised learning. This approach preserves the attractive properties of the decision tree while allowing the use of unlabeled samples. In particular, it is interpretable, easy to understand, quick to learn, and can handle numerical and nominal description characteristics. B. Jiang, et al. [6] proposed a novel graph-based semi-supervised learning framework which includes the sparse Bayesian semi-supervised learning method and the incremental sparse Bayesian semi-supervised learning method. The proposed algorithms can generate sparse solutions and make probabilistic prediction by transformation and induction. Label propagation algorithm (LPA) is a semi-supervised learning method based on graph. The labels are propagated from some known samples (vertices) to the unknown ones based on the similarities between the vertices in the graph [7]. LPA has the advantages of low complexity and high efficiency, and it has been widely applied in the fields of network community mining [8,9], information classification [10,11] and multimedia recognition and processing [12]. However, the traditional LPA simply takes the similarities between samples as the basis of label propagation, which lacks the evaluation criteria of new labeled samples. In addition, since the algorithm incorporates the new labeled samples into the training set, the propagation error may gradually increase, leading to the degeneration of the performance [13]. Due to the randomness during the process of propagation and the weak ability of LPA to deal with uncertain points, many new improved methods are proposed. C. Gong et al. [14] proposed a novel iterative LPA, in which each propagation alternates between two paradigms, teaching to learn and learning to teach (TLLT). In the ”teaching to learn” process, learners disseminate the simplest unmarked examples assigned by the teacher. In the ”learning-to-teach” step, the teacher adjusts the selection of the simplest subsequent example based on the learner’s feedback. J. Hao, et al. [15] adopted the fuzzy method when dealing with the categories of unlabeled samples. The categories of unlabeled samples are represented by the fuzzy membership degree in the interval of [0,1]. The final step of the algorithm is to remove the ambiguity. X. K. Zhang, et al. [16] dealt with the problem of random label selection by the value of node similarity. B. Wang et al. [17] proposed dynamic label propagation (DLP) to simultaneously deal with the multiclass and multi-label problem, the method in DLP is to update similarity measures dynamically by fusing multi-label and multi-class information.

In this paper, we put forward a Label Propagation based on Roll-back and Credibility (LPRC) algorithm to solve these problems. First, the credibility (label confidence) of each unlabeled sample is evaluated. According to the evaluation results, the label propagation order of the samples is determined, and the samples with high credibility are labeled in advance. Then, in order to avoid the impact of false label propagation, a roll-back mechanism based on feedback performance evaluation is proposed. In the process of label propagation in every fixed number of iterations, the new labeled samples will be default with the correct labels and added to the training set. At the same time, the samples in the original training set with the same size of new labeled samples will be chosen to moved out as a new dataset waiting to be labeled. The new labeled samples cannot be added to the training set until the propagation accuracy reached a predefined threshold, so we can reuse these new labeled samples in LPA with certain confidence.

The traditional LPA is described in section 2, and the details of LPRC algorithm is present in section 3. In section 4, we give the comparison results both on artificial synthetic dataset and UCI dataset, respectively. Finally, the conclusions and future research plan are given in section 5.

## 2. Label propagation algorithm

In label propagation algorithm, first, all categories in the sample set are required to be known. Assuming the number of categories in the samples set is t, $C = \{c_1, c_2, ..., c_t\}$ represents the collection of all the categories in the dataset. Then, the dataset is defined as follows: using $X_L = \{x_1, x_2, ..., x_l\}$ to represent the dataset of labeled samples, and $X_U = \{x_1, x_2, ..., x_u\}$ to represent the dataset of unlabeled samples. $X = X_L \cup X_U$, and $x_i \in \mathbb{R}^d$, $1 \leq l << u$. In addition, $Y_L = \{y_1, y_2, ..., y_l\}$ represents the labels set of labeled samples, $Y_U = \{y_1, y_2, ..., y_u\}$ represents the labels set of unlabeled samples. At the beginning, all labels of unlabeled samples can be set as 0 for the initial value of $Y_U$ is not so important.

The label propagation algorithm aims to spread labels of labeled samples to the samples with the greatest similarity. Take $X$ as the initial training matrix, classify the samples in $X_U$ through the labeled samples in $X_L$, add each round of newly labeled samples in $X_U$ to $X_L$, and then use the updated $X_L$ to conduct a new round of training on the unlabeled samples in $X_U$. The label propagation algorithm will repeat the above process until it reaches a certain end condition.

For a labeled sample $x_a \in X_L$, the similarity between it and the unlabeled sample $x_b \in X_U$ determines whether $x_a$ will spread its label to $x_b$. X. Zhu et al.[18] defines the similarity between any samples $x_i$ and $x_j$ as:

$$w_{ij} = exp(-\frac{d_{ij}^2}{\sigma^2}) = exp(-\frac{\sum_{d=1}^{D}(x_i^d - x_j^d)^2}{\sigma^2}) \tag{2.1}$$

The label of samples will be spread according to the similarity between samples themselves. Clearly, the greater the similarity between sample $x_a$ and $x_b$, the easier it is for $x_a$ to propagate its own label to $x_b$. In [18], a $(l + u) \times (l + u)$ probabilistic transition matrix $T$ is defined as (2.2), and $T_{ij}$ represents the possibility that node $i$ propagates its own label to node $j$.

$$T_{ij} = P(i \to j) = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}} \tag{2.2}$$

Meanwhile, a $(l + u) \times t$ label matrix $Y$ records the relationship between each sample and each category. $Y_{ij}$ represents the probability that the $i - th$ sample $x_i$ belongs to the $j - th$ class. The initial value of the unlabeled sample in the label matrix $Y$ is not very necessary to the classification result, and all the initial values in $Y$ can be set as 0. Using the probability propagation matrix $T$ to continuously update the label matrix $Y$, and then the maximum $s$ values in $Y$ are chosen. In other words, finding the $s$ unlabeled samples with the greatest similarity to the labeled samples, and assigning them to corresponding labels, then, moving them from $X_U$ to $X_L$. Repeat the process until the stop condition is satisfied.

## 3. LPRC algorithm

The traditional LPA only need a small number of labeled samples in the training process. In each subsequent training, the number of labeled sample dataset is continuously expanded by using the newly labeled samples to improve the accuracy of classification. But in this process, there are also some problems that we cannot ignore. In reality, the differences between samples belong to different categories are often not so clear, and the samples at the edge of the category are sparser than those at the center of the category.

As shown in Figure 1(a), the samples on the boundary of two categories are identified as key nodes. If the key nodes are wrongly labeled during the label propagation procedure, it would bring serious impacts on the latter label propagation of unknown samples. Just as Figure 1(b) shows.

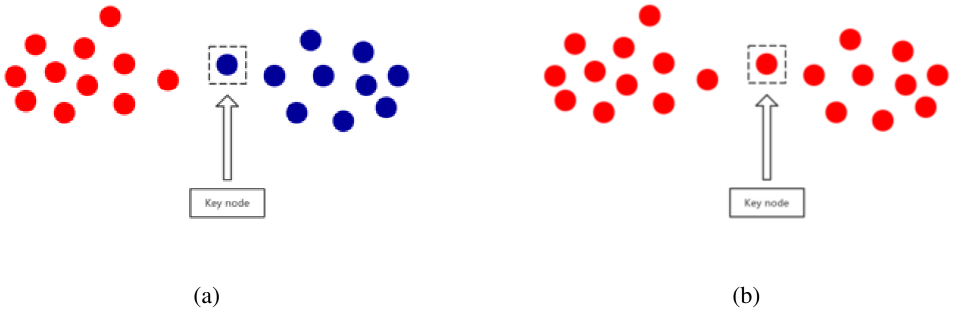


**Figure 1.** A distribution assumed to explain the importance of key node. (a) The samples with correct classes; (b) The result of key node being misclassified.

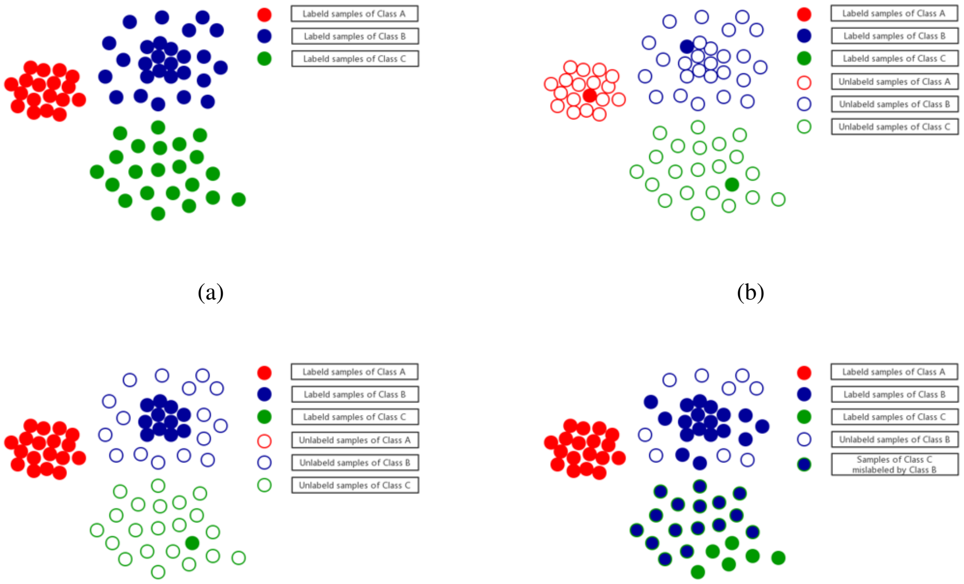
Therefore, we propose LPRC algorithm. First, we will evaluate the credibility of these unlabeled samples, instead of classifying them directly, and the sequence of classification operation of unlabeled samples is determined according to the credibility value of themselves. For those samples with high credibility values, will be classified first. This part will be described in detail in section 3.2.

In addition, considering the distribution differences of samples in each category in the dataset, we also put forward new ideas on how to select samples for labeling in each round. And this part will be described in section 3.1.

Finally, as the label propagation algorithm constantly updates the training matrix and increases the number of labeled samples, the classification error may be continuously expanded, resulting in a drastically reduced accuracy. Therefore, we propose a roll-back mechanism based on feedback effect detection. In the process of label propagation, the algorithm performs feedback accuracy detection on the currently labeled samples every k rounds of iteration. Only when the set conditions are satisfied, the algorithm will continue the next round of iteration. Otherwise, the samples labeled in this round will be discarded and the label propagation of this round will be carried out again. This part will be described in detail in section 3.2.

### 3.1. The selection of unlabeled samples

In a dataset, we can find that the distribution of samples in different categories is often different, which will also have a certain impact on the label propagation algorithm. Figure 2(a) shows a distribution of the samples in a dataset. We can see that the samples of class A are more densely distributed than those of the other two categories. The distribution of class B samples gradually becomes sparse from the center. The distribution of class C samples is relatively uniform. There is an obvious distance between each pair of adjacent samples in Class C, which means C is sparser than the other two categories.


Labeld samples of Class A
Labeld samples of Class B
Labeld samples of Class C
Labeld samples of Class A
Labeld samples of Class B
Labeld samples of Class C
Unlabeld samples of Class A
Unlabeld samples of Class B
Unlabeld samples of Class C
Labeld samples of Class A
Labeld samples of Class B
Labeld samples of Class C
Unlabeld samples of Class A
Unlabeld samples of Class B
Unlabeld samples of Class C
Labeld samples of Class A
Labeld samples of Class B
Labeld samples of Class C
Unlabeld samples of Class B
Samples of Class C mislabeled by Class B

(a)

(b)

(c)

(d)

**Figure 2.** (a) A dataset with all labeled samples. 2(b)–2(d) A propagating step of label propagation. (b) The initial state with only 3 labeled samples. (c) A situation that may occur in the course of propagation in one round. (d) The result because of the different density of the sample distribution.

According to the label propagation algorithm, each round will spread corresponding labels to s unlabeled samples that are most similar to the labeled samples. At this time, since the samples of class C are more sparsely distributed than those of the other two classes, the similarity $w_c$ between any two samples $x_{c_1}$ and $x_{c_2}$ of class C is generally lower than $w_a$ ($w_b$) between $w_{a_1}$ and $w_{a_2}$ of any two samples of class A (class B). This may lead to that class C has not received any labels spread from the labeled samples of the class during the first n iterations.

However, the labeled sample dataset of the other two classes may misclassify the samples on the edge of the category in the subsequent propagation due to the continuous expansion. As a result, with the constant updating of $X_L$, the classification errors in the later stage will become larger and larger, and even the phenomenon of class A (class B) merging the samples of C may occur. Figure 2(b)–2(d) have shown this situation. Although the samples of class B in the dashed box are at the edge, their distribution is sparser than that of the samples in the center of the category. Compared with the samples of class C, whose samples are evenly and sparsely distributed in the original category, they are easier to be spread corresponding labels by the similar samples. So, as shown in figure 2(d), the samples in the solid box that originally belong to class C may be wrongly classified as class B. With iteration after iteration of the algorithm, this error will be gradually amplified, and resulting in a large number of these samples being swallowed by class B.

In order to avoid the situation mentioned above, we no longer select the s most similar samples from the entire unlabeled sample dataset, but from the perspective of each category to consider them separately when we select unlabeled samples to label in each round. That is, $\beta = s/t$ unlabeled samples most similar to the labeled samples of the class are selected from the $t$ classes for labeling respectively. In each round of iteration, new labeled samples are guaranteed to be added to each category, so that the labeled sample dataset of each class can be uniformly expanded, avoiding the phenomenon of rapid expansion of a certain class and large disparity in volume between categories, thus reducing or even avoiding category annexation.

At the later stage of the algorithm iteration, the value of s may need to be recalculated. 1. All unlabeled samples of $t_1$ certain classes are classified over: set $t = t - t_1$, $s = \beta \times t$. 2. The number of unlabeled samples $\alpha$ remaining in each class exists $\alpha < \beta$: set $\beta = \alpha$, $s = \beta \times t$.

### *3.2. Credibility assessment*

As previously mentioned in section 2, the traditional label propagation algorithm obtains the largest $s$ values directly from the label matrix $Y$, and continuously updates $Y$ by using the probability propagation matrix $T$. Namely to find the s unlabeled samples with the highest similarity to the labeled samples, assign them to the corresponding labels, and move them from $X_U$ to $X_L$.

However, such a selection mechanism actually only considers one aspect of ”similarity”. At this point, we need to conduct a classification credibility assessment of these unlabeled samples.

When we get a labeled sample data set, we often hope that these samples can well represent the classes to which they belong, and there are obvious differences between them. It means that these samples are distributed in the center of their respective categories. In practice, however, we cannot guarantee the quality of the labeled samples we get. For example, most of these samples are on the boundary of two adjacent classes, so it is likely that the distance between these samples is not as significant as we expected. If a label propagation algorithm is used at this time, there may be cases where the similarity of some unlabeled samples to two (or more) labeled samples of different classes is close to and higher than all other similarities, which means the probability that the unlabeled sample may be assigned to its own label by two (or more) different categories is similar. The reason is that the label propagation generated by traditional methods is completely controlled by the adjacency relationship between samples, including those labeled and unlabeled samples. The labels of labeled samples are blindly spread to unlabeled adjacent samples without considering the difficulty and risk of transmission [14].

As the labeled sample dataset continues to expand with the iteration of the algorithm, we hope that the quality of this dataset can be guaranteed. In this way, the classification accuracy will not be significantly reduced, and the probability of label propagation between samples of different categories can be controlled. Therefore, we need to make an assessment of the label propagation order of unlabeled samples.

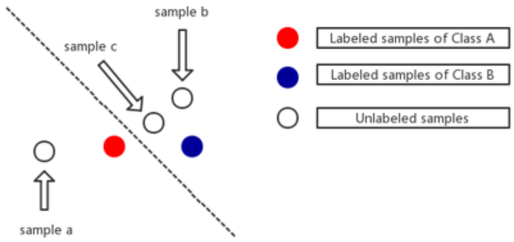


**Figure 3.** A distribution of unlabeled samples, the existing labeled samples are shown as the red and blue solid circles.

As shown in Figure 3, we assume that there is such a sample distribution: the dividing line of A and B is indicated by the dotted line in the figure, and the existing labeled samples are shown as the red and blue solid circles in the figure. In this round of update of the algorithm, there are samples a-c waiting to be labeled. The distance $d_{Ac}$ from sample c to class A is less than the distance $d_{Aa}$ from sample a to class A, the distance $d_{Bc}$ from sample c to class B is less than the distance $d_{Bb}$ from sample b to class B, and here we have: $d_{Ac} - d_{Bc} = \lambda, \lambda \rightarrow 0$.

So, the probability $P_{Ac}$ that sample c belongs to class A, and the probability $P_{Bc}$ that sample c belongs to class B are both large and they are very close. In the traditional label propagation algorithm, the label will be assigned to sample c according to $max(P_{Ac}, P_{Bc})$, but this may cause the wrong classification of sample c.

In our newly proposed LPRC algorithm, we will further process the label matrix Y after each iteration under the conditions in section 3.1: sorting the label sequence of samples, so as to improve the classification accuracy of the label propagation algorithm.

As we mentioned in Section 2, in the label matrix $Y$, $Y_{ij}$ represents the probability that the $i-th$ sample belongs to the $j-th$ class. Then, when the situation we discussed in this section occurs, it means the set of categories for the dataset $C = \{c_1, c_2, ..., c_t\}$, $\exists x_i \in X_U, Y_{x_ic_1} - Y_{x_ic_2} \leq \lambda, \lambda \rightarrow 0$. At this point, for any sample $x_i \in X_U$, there is a set of label vectors $Y_{x_iC} = \{Y_{x_ic_1}, Y_{x_ic_2}, ..., Y_{x_ic_t}\}$, represents the probability that the sample $x_i$ belongs to each category. We find out the maximum value $Y_{x_ic_n}$ and the second largest value $Y_{x_ic_m}$ from the set of label vectors, and set $D_{value} = Y_{x_ic_n} - Y_{x_ic_m}$.

Here we can get: $Y = \begin{bmatrix} Y_{x_1c_1} & Y_{x_1c_2} & \cdots & Y_{x_1c_t} \\ Y_{x_2c_1} & Y_{x_2c_2} & \cdots & Y_{x_2c_t} \\ \cdots & \cdots & \cdots & \cdots \\ Y_{x_uc_1} & Y_{x_uc_2} & \cdots & Y_{x_uc_t} \end{bmatrix}$

It exists in $Y_{x_iC} = \{Y_{x_ic_1}, Y_{x_ic_2}, ..., Y_{x_ic_t}\}$: $Y_{x_ic_n} = D_{max_i} = max(Y_{x_ic_1}, Y_{x_ic_2}, ..., Y_{x_ic_t})$, and $DM_{max_i} = max(Y_{x_ic_1}, Y_{x_ic_2}, ..., Y_{x_ic(n-1)}, Y_{x_ic(n+1)}, ..., Y_{x_ic_t})$. Moreover, here we need to use $Class = \{class_1, class_2, ..., class_u\} \in C$ to record the category $C_n$ indicated by $Y_{x_ic_n}$. Setting $D_{vp_i} = (D_{value_i}, class_i)$, then we can get a two-dimensional matrix $D_{vp}$ to store the value of $D_{value}$, and $class_n$ represents the class $C_n$ with the highest probability that the sample belongs to: $D_{vp} =$

$$\begin{bmatrix} D_{max_1} - DM_{max_1} & class_1 \\ D_{max_2} - DM_{max_2} & class_2 \\ ... & ... \\ D_{max_u} - DM_{max_u} & class_u \end{bmatrix}$$

In each round of update of label matrix $Y$, the sample set is first divided according to the category set $C$ and the value of the sample in set $Class$. Samples have the same value in set Class will be divided into a subset. Select $\beta$ samples with the largest values in $D_value$ from each subset to label. That is, $\beta$ unlabeled samples are selected from $t$ categories each time should be sure that, the probability that these unlabeled samples belong not only to the certain class is as large as possible, but also to other classes is as small as possible. This operation could ensure the credibility of samples added to the labeled sample dataset in each round. We refer to such samples as easy-to-label samples.

If the samples a-c in Figure 4 is evaluated here based on $D_value$ and $Class$, the sequence of them to be labeled can be changed.

The label matrix $Y$ composed of the label vector set of samples a–c is: $Y = \begin{bmatrix} Y_aA & Y_aB \\ Y_bA & Y_bB \\ Y_cA & Y_cB \end{bmatrix}$

After calculating its $D_{value}$ and $Class$, respectively, we can get: $D_{vp} = \begin{bmatrix} D_{max_a} - DM_{max_a} & class_1 \\ D_{max_b} - DM_{max_b} & class_2 \\ D_{max_c} - DM_{max_c} & class_3 \end{bmatrix}$

According to Figure 5, for sample c: $Y_{cA} - Y_{cB} = \lambda, \lambda \rightarrow 0$, therefore, it will be labeled at the end.

Compared with sample c, sample a and b are not only close to their own classes, but also keep a relatively obvious distance from the different classes to some extent. Such samples like a and b are more reliable in classification.

After samples a and b are given corresponding labels (A and B) respectively, sample c will be affected by both the original labeled samples and the newly labeled samples a and b. We will get the result that sample c is divided into class B by superimposing these effects before, which is exactly in line with the actual distribution of the sample dataset.

### *3.3. Roll-back detection mechanism*

The advantage of the label propagation algorithm is that it is simple, fast and efficient, but it also has the drawback that the results of each iteration are unstable and the accuracy is not high. In order to control this instability, we added a detection mechanism in the process of algorithm iteration, which called ”roll-back”.

As we know, the traditional label propagation algorithm will label $s$ new samples in each round of iteration and add this part of the samples to the labeled sample dataset. If there is a classification error in the newly added sample in this part, then the classification errors in the later stage will increase with the iteration of the algorithm. So here we try to use the ”roll-back” detection mechanism to control this error.

After each round of iteration, we will get s samples of new labels. At this time, we add a condition: each iteration of $k$ rounds, a roll-back detection is carried out for the newly labeled $s$ samples of the round. Only when the condition is satisfied, the algorithm continues to execute. Otherwise, the newly labeled s samples of the round will be discarded and the label propagation of the round will be carried out again. Under the conditions in section 3.1, the flow path of the roll-back detection mechanism set for the new labeled samples of the round is as follows:

1. Use $N_s^k$ to represent the dataset composed of $s$ unlabeled samples with new labels in the $k-th$ round, and add $N_s^k = \{x_{n_1}, x_{n_2}, ..., x_{n_s}\} \in X_U$ to $X_L$. By default, all labels owned by samples in $N_s^k$ are correct.

2. Find the center sample of $N_s^k$ and then calculate the distance $Dis = \{D_1, D_2, ..., D_S\}$ between each sample and the center sample. Draw a circle with $max(Dis)$ as the radius $r$, search for samples in $X_L^k$ within this range, and remove them out as $U_s$. If the sample number within this range is $s_1$, and $s_1 < s$, we will randomly select $s - s_1$ samples from $X_L^k - U_{s_1}$ to $U_s$.

3. Use $X_L^k$ to represent the labeled sample dataset used for training in the $k-th$ round, select $s$ samples to be removed out from $X_L^k$ as a new independent unlabeled sample dataset $U_s$.

4. Let a conduct label propagation on samples in $U_s$, and detect the accuracy of newly transmitted labels on samples in $U_s$.

5. Set the accuracy rate as $R_p$ and the threshold as $\eta$. If $R_p < \eta$, then the samples in $N_s^k$ will be discarded and the round of label propagation will be carried out again. If $R_p < \eta$, the algorithm continues.

## 4. Experimental results and analysis

In order to evaluate the effectiveness of the method proposed in this paper, we conducted experiments in the artificial synthetic dataset and UCI dataset, respectively. The hardware environment of the experiment is: Intel(R) Core(TM) i7-9700, CPU 3.00GHz, and the software environment is: Windows10+Matlab2017a+Python3.6.

### *4.1. Synthetic dataset*

In Python, we used the method (sklearn.datasets()) provided by the sklearn library to generate the 5 synthetic datasets [19] we needed, as shown in figure 4 and table 1. Dataset circles is a double ring shape, and it contains 1500 samples and 2 classes, 750 samples in each class. Dataset moons is made by 1500 samples and it also contains 2 classes, 750 samples in each class. Dataset varied, aniso, blobs each has 1500 samples and 3 classes, 500 samples in each class.

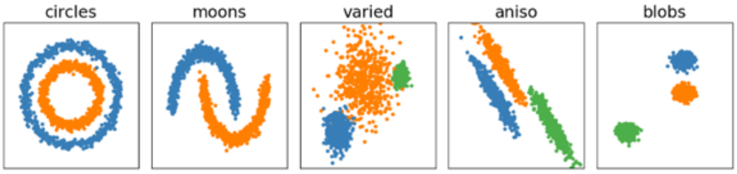


**Figure 4.** The illustration of artificial synthetic datasets, to show the distribution of samples.

We used 9 algorithms and LPRC algorithm for experimental comparison. These algorithms have been proved to have high classification performance, and have been applied in many fields. And they can adapt well to the situation of only a small number of labeled samples is known. The classification effects of the 10 algorithms on the five artificially synthesized datasets and the original datasets are shown in Figure 5, and the specific classification accuracy is shown in table 2. Labeled rate represents the percentage of labeled samples in each dataset.

LPRC (labeled rate = 0.01,s = 15), KNN (labeled rate = 0.01,k = 1) [20], DecisionTreeClassifier (labeled rate = 0.01) [21], GaussianNB (labeled rate = 0.01) [22], LPA(label rate = 0.01, s = 15) and keep the number of labeled samples for each category has the same proportion as the number of samples for that category.

**Table 1.** The detail of each artificial synthetic dataset.

| Datasets | Sum | Classes | Sum of each class |
|---|---|---|---|
| Circles | 1500 | 2 | 750 |
| Moons | 1500 | 2 | 750 |
| Varied | 1500 | 3 | 500 |
| Aniso | 1500 | 3 | 500 |
| Blobs | 1500 | 3 | 500 |

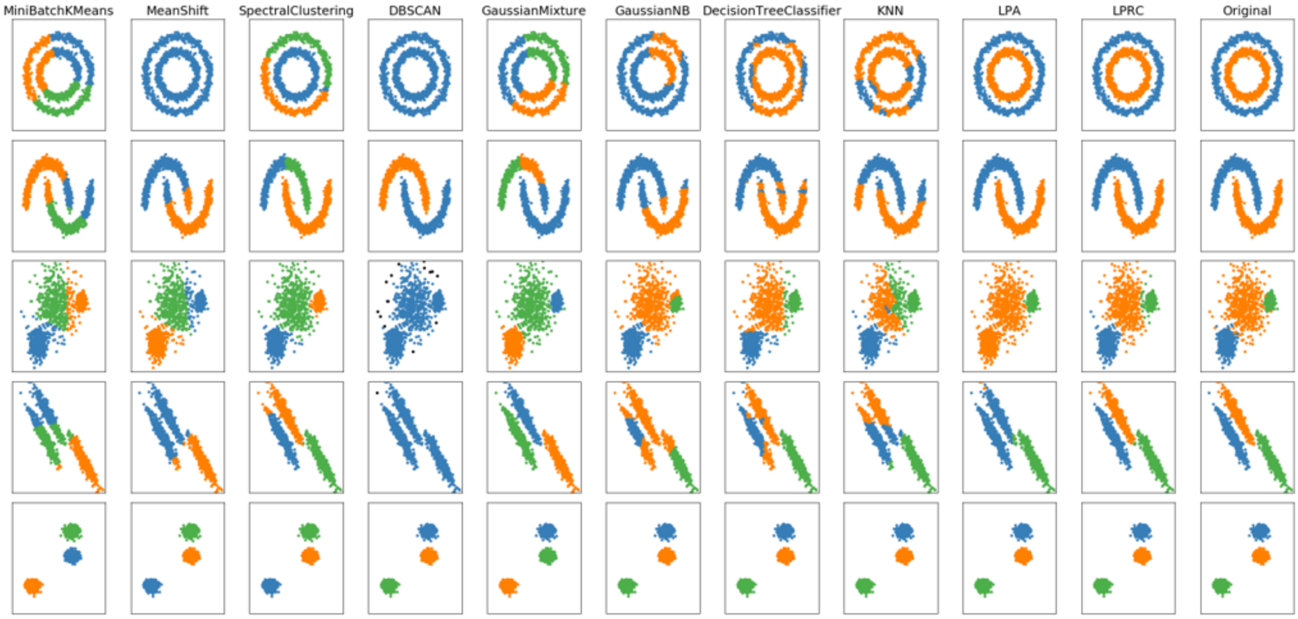


**Figure 5.** The classification result of the 10 algorithms on 5 synthetic datasets, the last column shows the original datasets.

**Table 2.** The accuracy of the 10 algorithms on 5 synthetic datasets.

| Datasets<br>Algorithms | Circles | Moons | Varied | Aniso | Blobs |
|---|---|---|---|---|---|
| MiniBatchKmeans | 33 % | 17 % | 38 % | 7 % | 0 % |
| Meanshift | 50 % | 87 % | 2 % | 32 % | 33 % |
| SpectralClustering | 1% | 74 % | 34 % | 99 % | 33 % |
| DBSCAN | 50 % | 0 % | 33 % | 33 % | 100 % |
| GaussianMixture | 33 % | 9 % | 1 % | 0 % | 33 % |
| GaussianNB | 58.48 % | 86.34 % | 93.93 % | 79.26 % | 99.86 % |
| DecisionTree | 75.77 % | 89.16 % | 95.42 % | 90.37 % | 100 % |
| KNN | 62.25 % | 82.78 % | 89.02 % | 87.88 % | 100 % |
| LPA | 99.86 % | 99.93 % | 66.53 % | 67.80 % | 100 % |
| LPRC | 100 % | 100 % | 97.80 % | 99.80 % | 100 % |

By comparing the classification effects of the above 10 algorithms on 5 synthetic datasets, we can see that the classification ability of LPRC algorithm is better than other algorithms on the whole, and it can also better match the distribution of samples.

### *4.2. UCI dataset*

The datasets iris, wine, australian, breast, downloaded through UCI [23], also used the above 10 algorithms for experimental comparison. The information for the data set is shown in table 3. Dataset iris has 150 samples and 3 classes, and the number of features is 4. Dataset wine contains 178 samples for 3 classes, and 13 features. Dataset australian contains 690 samples and 14 features, and dataset breast contains 699 samples and 10 features. Both of dataset australian and breast has 2 classes.

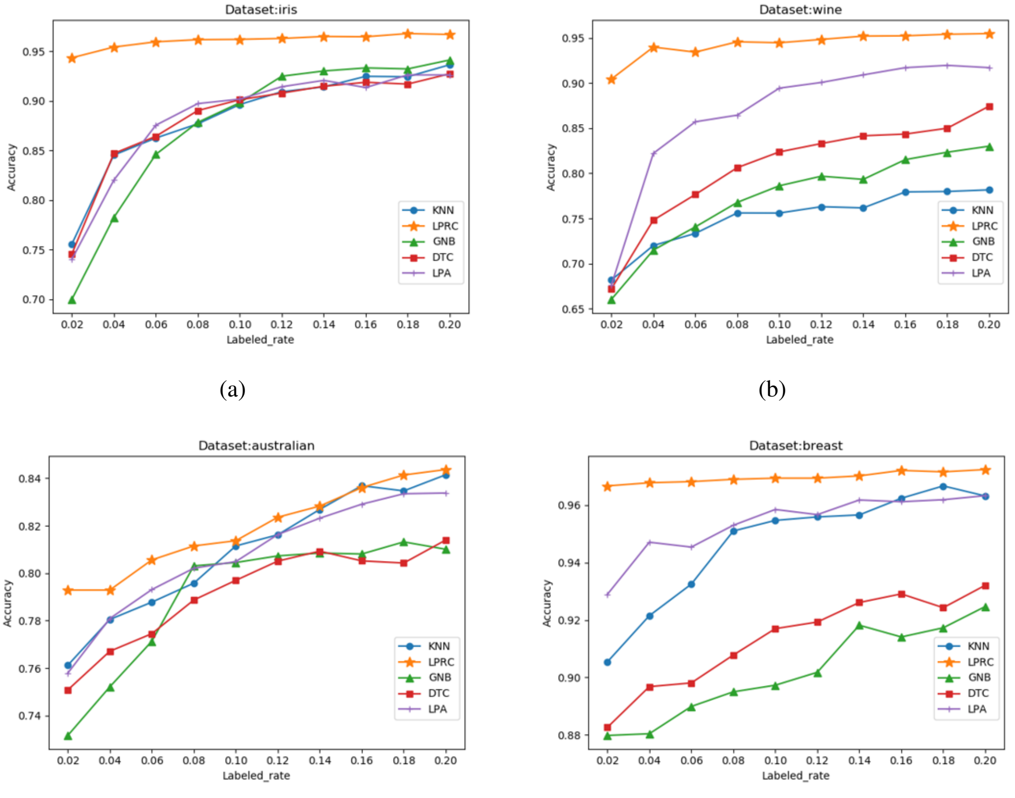
The classification results of 5 algorithms, LPRC, KNN, GaussianNB (GNB), DecisionTreeClassifier (DTC) and LPA with different labeled rate are shown in figure 6(a)-(d) and table 5 (average value of 50 experiments is taken for each result), and the classification results of the other 5 algorithms are shown in table 4.

**Table 3.** The details of 4 UCI datasets.

| Datasets | Sum of samples | Number of features | Number of classes |
|---|---|---|---|
| Iris | 150 | 4 | 3 |
| Wine | 178 | 13 | 3 |
| Australian | 690 | 14 | 2 |
| Breast | 699 | 14 | 2 |

**Table 4.** Accuracies of other 5 algorithms on these 4 UCI datasets.

| Dataset / Algorithm | Iris | Wine | Australian | Breast |
|---|---|---|---|---|
| MiniBatchKmeans | 2% | 10% | 47% | 6% |
| Meanshift | 33% | 7% | 49% | 4% |
| SpectralClustering | 17% | 16% | 32% | 40% |
| DBSCAN | 17% | 4% | 52% | 3% |
| GaussianMixture | 0% | 18% | 44% | 9% |



(a)

(b)

(c)

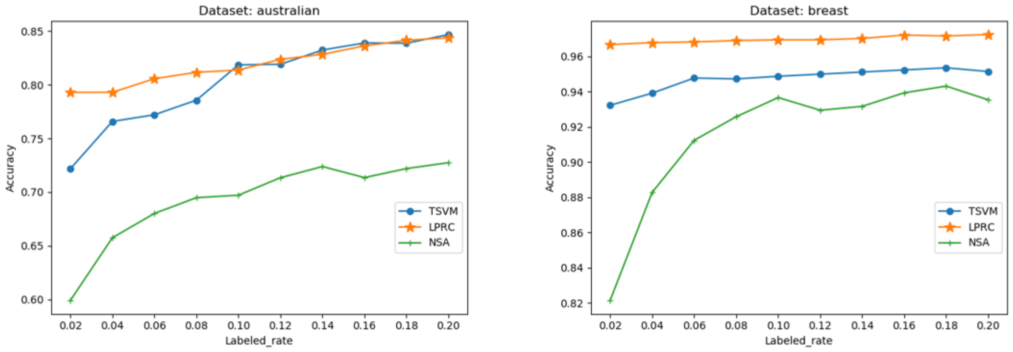
(d)

**Figure 6.** The recognition accuracies with different labeled rate for different algorithms. (a) The classification accuracies of these algorithms with different labeled rate on dataset wine. (b) The results on dataset iris. (c) The results on dataset australian. (d) The results on dataset breast. Compared with these experimental results can prove the good performance in classification of LPRC, especially when the labeled rate is small.

Otherwise, an additional experiment is made to compare the performance of LPRC with TSVM [24] and negative selection algorithm (NSA), these algorithms just need a small number of labeled samples. The results on UCI datasets of these 3 algorithms are shown in figure 7(a)–(b) and table 6, and the results on synthetic datasets are shown in figure 8.

**Table 5.** Accuracies with different labeled rate of the 5 algorithms on these 4 UCI datasets.

| Dataset | Algorithm<br>labeled rate | GaussianNB | DecisionTree | KNN | LPA | LPRC |
|---|---|---|---|---|---|---|
| iris | 2% | 69.96% | 74.54% | 75.54% | 74.00% | 94.33% |
| | 4% | 78.22% | 84.68% | 84.82% | 81.99% | 95.41% |
| | 6% | 84.59% | 86.41% | 86.24% | 87.53% | 95.94% |
| | 8% | 87.82% | 89.01% | 87.67% | 89.71% | 96.16% |
| | 10% | 89.80% | 90.10% | 89.62% | 90.16% | 96.19% |
| | 12% | 92.47% | 90.77% | 90.91% | 91.41% | 96.28% |
| | 14% | 93.00% | 91.46% | 91.40% | 92.05% | 96.48% |
| | 16% | 93.31% | 91.86% | 92.46% | 91.33% | 96.45% |
| | 18% | 93.21% | 91.68% | 92.42% | 92.60% | 96.73% |
| | 20% | 94.11% | 92.72% | 93.63% | 92.60% | 96.68% |
| wine | 2% | 66.02% | 67.19% | 68.17% | 67.58% | 90.44% |
| | 4% | 71.49% | 74.81% | 72.00% | 82.21% | 93.99% |
| | 6% | 74.06% | 77.65% | 73.34% | 85.71% | 93.43% |
| | 8% | 76.78% | 80.63% | 75.61% | 86.45% | 94.58% |
| | 10% | 78.62% | 82.37% | 75.60% | 89.44% | 94.47% |
| | 12% | 79.68% | 83.30% | 76.30% | 90.07% | 94.84% |
| | 14% | 79.33% | 84.16% | 76.17% | 90.91% | 95.21% |
| | 16% | 81.52% | 84.35% | 77.94% | 91.71% | 95.24% |
| | 18% | 82.34% | 85.00% | 77.99% | 91.97% | 95.41% |
| | 20% | 83.01% | 87.45% | 78.17% | 91.71% | 95.49% |
| australian | 2% | 73.16% | 75.08% | 76.12% | 75.79% | 79.29% |
| | 4% | 75.21% | 76.71% | 78.05% | 78.09% | 79.29% |
| | 6% | 77.13% | 77.45% | 78.78% | 79.31% | 80.56% |
| | 8% | 80.31% | 78.87% | 79.58% | 80.21% | 81.15% |
| | 10% | 80.45% | 79.70% | 81.15% | 80.49% | 81.37% |
| | 12% | 80.73% | 80.51% | 81.62% | 81.63% | 82.36% |
| | 14% | 80.85% | 80.93% | 82.69% | 82.32% | 82.83% |
| | 16% | 80.81% | 80.52% | 83.69% | 82.91% | 83.61% |
| | 18% | 81.32% | 80.43% | 83.47% | 83.35% | 84.13% |
| | 20% | 81.01% | 81.40% | 84.15% | 83.38% | 84.37% |
| breast | 2% | 87.98% | 88.26% | 90.54% | 92.89% | 96.67% |
| | 4% | 88.04% | 89.68% | 92.15% | 94.71% | 96.78% |
| | 6% | 88.99% | 89.81% | 93.25% | 94.54% | 96.82% |
| | 8% | 89.50% | 90.78% | 95.10% | 95.30% | 96.90% |
| | 10% | 89.73% | 91.70% | 95.47% | 95.85% | 96.94% |
| | 12% | 90.18% | 91.93% | 95.59% | 95.67% | 96.94% |
| | 14% | 91.82% | 92.61% | 95.66% | 96.18% | 97.02% |
| | 16% | 91.41% | 92.91% | 96.24% | 96.12% | 97.21% |
| | 18% | 91.73% | 92.43% | 96.67% | 96.19% | 97.16% |
| | 20% | 92.47% | 93.21% | 96.32% | 96.33% | 97.24% |



(a) (b)

**Figure 7.** The recognition accuracies with different labeled rate for different algorithms. (a) The classification accuracies of these algorithms with different labeled rate on dataset australian. (b) The results on dataset breast.

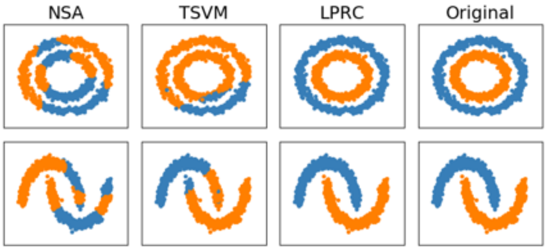


**Figure 8.** The classification result of the 3 algorithms on 2 synthetic datasets.

After experimental comparison, we found that the LPRC algorithm has good performance in classification, and it especially has high accuracy when only has a small number of labeled samples. As the number of labeled samples increases, the accuracy rate also shows an increasing trend. In the process of propagation, the error caused by the addition of new labeled samples is reduced, and the case of one class of samples being swallowed by another is prevented. The classification effect has been significantly improved.

## 5. Statistical test for comparison of LPRC

Another experiment is made to further test the method LPRC we proposed. The result is analyzed by a significance test to assess the effectiveness of LPRC. Table 7 shows the average results for 50 runs of LPA and LPRC on 5 UCI datasets, and we can find that the accuracies of LPRC are higher than LPA. The statistical test of the results is based on two hypotheses of the average accuracy acc values of LPRC, where $u_0$ is the average accuracy of LPA:

$$\begin{cases} H_0 : acc \text{ is similar with } u_0 \\ H_1 : acc \text{ is significantly bigger than } u_0; \end{cases}$$

Based on the central limit theorem, the average accuracy obtained by repeating the algorithm can

**Table 6.** Accuracies with different labeled rate of the 3 algorithms on these 2 UCI datasets.

| Dataset | Algorithm<br>labeled rate | TSVM | NSA | LPRC |
|---|---|---|---|---|
| australian | 2% | 72.17% | 59.91% | 79.29% |
| | 4% | 76.58% | 65.75% | 79.29% |
| | 6% | 77.20% | 68.01% | 80.56% |
| | 8% | 78.56% | 69.48% | 81.15% |
| | 10% | 81.85% | 69.71% | 81.37% |
| | 12% | 81.90% | 71.34% | 82.36% |
| | 14% | 83.23% | 72.38% | 82.83% |
| | 16% | 83.89% | 71.35% | 83.61% |
| | 18% | 83.86% | 72.19% | 84.13% |
| | 20% | 84.67% | 72.74% | 84.37% |
| breast | 2% | 93.22% | 82.13% | 96.67% |
| | 4% | 93.91% | 88.27% | 96.78% |
| | 6% | 94.77% | 91.23% | 96.82% |
| | 8% | 94.72% | 92.57% | 96.90% |
| | 10% | 94.87% | 93.66% | 96.94% |
| | 12% | 94.99% | 92.94% | 96.94% |
| | 14% | 95.11% | 93.16% | 97.02% |
| | 16% | 95.23% | 93.93% | 97.21% |
| | 18% | 95.35% | 94.31% | 97.16% |
| | 20% | 95.14% | 93.53% | 97.24% |

**Table 7.** The results of statistical test ($\alpha = 0.05$).

| Dataset | Algorithm<br>labeled rate | LPA | | LPRC | | Statistical test |
|---|---|---|---|---|---|---|
| | | $acc_1$ | $var_1$ | $acc_2$ | $var_2$ | |
| iris | 2% | 74.00% | 1.03E-02 | 94.33% | 1.74E-03 | 138.17 |
| | 4% | 81.99% | 7.25E-03 | 95.41% | 7.26E-04 | 129.57 |
| | 6% | 87.53% | 6.59E-03 | 95.94% | 4.59E-05 | 89.33 |
| | 8% | 89.71% | 1.96E-03 | 96.16% | 4.01E-05 | 230.36 |
| | 10% | 90.16% | 1.67E-03 | 96.19% | 4.27E-05 | 252.75 |
| | 12% | 91.41% | 1.22E-03 | 96.28% | 4.99E-05 | 279.43 |
| | 14% | 92.05% | 9.17E-04 | 96.48% | 4.27E-05 | 338.17 |
| | 16% | 91.33% | 8.02E-04 | 96.45% | 5.41E-05 | 446.88 |
| | 18% | 92.60% | 5.07E-04 | 96.73% | 4.13E-05 | 570.21 |
| | 20% | 92.60% | 4.72E-04 | 96.68% | 5.60E-05 | 605.09 |
| wine | 2% | 67.58% | 1.67E-02 | 90.44% | 7.64E-03 | 95.82 |
| | 4% | 82.21% | 1.17E-02 | 93.99% | 4.30E-04 | 70.47 |
| | 6% | 85.71% | 6.44E-03 | 93.43% | 1.65E-03 | 83.91 |
| | 8% | 86.45% | 5.85E-03 | 94.58% | 3.50E-05 | 97.28 |
| | 10% | 89.44% | 1.63E-03 | 94.47% | 2.10E-04 | 216.01 |
| | 12% | 90.07% | 1.28E-03 | 94.84% | 6.53E-05 | 260.86 |
| | 14% | 90.91% | 1.33E-03 | 95.21% | 7.35E-05 | 226.32 |
| | 16% | 91.71% | 7.66E-04 | 95.24% | 4.83E-05 | 319.84 |
| | 18% | 91.97% | 7.70E-04 | 95.41% | 5.98E-05 | 312.72 |
| | 20% | 91.71% | 5.85E-04 | 95.49% | 6.25E-05 | 452.31 |
| australian | 2% | 75.79% | 5.46E-03 | 79.29% | 3.12E-03 | 44.87 |
| | 4% | 78.09% | 4.64E-03 | 79.29% | 3.79E-03 | 18.10 |
| | 6% | 79.31% | 3.51E-03 | 80.56% | 1.38E-03 | 24.93 |
| | 8% | 80.21% | 2.72E-03 | 81.15% | 1.17E-03 | 24.18 |
| | 10% | 80.49% | 1.26E-03 | 81.37% | 1.16E-03 | 48.89 |
| | 12% | 81.63% | 1.36E-03 | 82.36% | 9.49E-04 | 37.57 |
| | 14% | 82.32% | 6.70E-04 | 82.83% | 6.64E-04 | 53.28 |
| | 16% | 82.91% | 5.16E-04 | 83.61% | 5.95E-04 | 94.96 |
| | 18% | 83.35% | 6.27E-04 | 84.13% | 4.27E-04 | 87.08 |
| | 20% | 83.38% | 6.11E-04 | 84.37% | 5.08E-04 | 113.42 |
| breast | 2% | 92.89% | 2.12E-03 | 96.67% | 8.78E-06 | 124.80 |
| | 4% | 94.71% | 4.79E-04 | 96.78% | 1.31E-05 | 302.50 |
| | 6% | 94.54% | 7.34E-04 | 96.82% | 1.23E-05 | 217.44 |
| | 8% | 95.30% | 1.68E-04 | 96.90% | 1.05E-05 | 666.66 |
| | 10% | 95.85% | 8.29E-05 | 96.94% | 1.11E-05 | 920.39 |
| | 12% | 95.67% | 1.14E-04 | 96.94% | 1.70E-05 | 779.82 |
| | 14% | 96.18% | 9.49E-05 | 97.02% | 2.14E-05 | 619.60 |
| | 16% | 96.12% | 8.01E-05 | 97.21% | 1.79E-05 | 952.55 |
| | 18% | 96.19% | 1.17E-04 | 97.16% | 2.04E-05 | 580.35 |
| | 20% | 96.33% | 5.68E-05 | 97.24% | 1.87E-05 | 1121.48 |

be assumed to follow a normal distribution. According to [25], $\frac{(acc-u_0)}{(s/\sqrt{n})}$ coincides with $T(n-1)$, and if $H_0$ established, the average accuracy $acc$ would be close to the value of $u_0$ for $H_0$. Otherwise, $H_0$ would be rejected with a confidence level of $1-\alpha$ when $\frac{(acc-u_0)}{(s/\sqrt{n})} \geq T_\alpha(n-1)$ is satisfied. We use $s$ to represent the sample variation and $n$ is the number of repetitions.

The results of the statistical test are shown in table 7, $acc_1$ and $acc_2$ represent the average accuracies of LPA and LPRC, respectively.

As can be observed in table 7, for the given confidence level, all the test results are higher than the rejection threshold $T_{\alpha=0.05}(49) = 1.6777$. It means that H0 does not established and $H_1$ is true. This experiment proved that the proposed method LPRC is effective.

## 6. Conclusion and future work

In this paper, a new algorithm LPRC is proposed to improve the stability of the traditional LPA. To achieve better propagation results, a credibility assessment and a roll-back detection schemes are designed. The credibility assessment of each sample is calculated first to determinate the label propagation order, which ensures that the new labeled samples are more reliable to be added to the labeled set for future propagation. Then, a roll-back mechanism based on feedback detection is used to the control the propagation error caused by wrong labels. Only when the exit conditions are satisfied, the new labeled samples could maintain their labels and be moved to labeled sample dataset, or the new labeled samples in this round will be discarded.

LPRC not only maintains the original simple and efficient features of label propagation, but also increases its accuracy in classification. The comparisons based on the artificial synthetic datasets and the UCI datasets demonstrated that classification performance of LPRC are obviously better than traditional algorithms. In particular, it is suitable to the situation with only a small number of labeled samples.

In the feature, we will continue to make deep research in label propagation algorithm in order to let it exert the best performance. The research we made is just based on the static samples, but in the practice, the samples are always dynamical. Considering with this situation, we will focus on the dynamic samples in the next step to fit the practical applications better.

## Acknowledgments

This work was supported by the Natural Science Foundation of China (Grantnos.61872255, U1736212 and U19A2068).

## Conflict of interest

The authors declare there is no conflict of interest.

References

1. Z. H. Zhou, M. Li, Semi-supervised learning by disagreement, *Knowl. Inf. Syst.*, **24** (2010), 415–439.

2. X. Zhu, Z. Ghahramani, J. D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, *Proc. 20th. Int. Conf. Mach Learn.(ICML)*, Washington DC, USA, August, **2003** (2003), 912–919.

3. O. Chapelle, B. Scholkopf, A. Zien, Semi-supervised learning, *IEEE Trans. Neural Netw. Learn. Syst.*, **20** (2006), 542–542.

4. K. P. Bennett, A. Demiriz, Semi-supervised support vector machines, *Adv. Neural Inform. Proc. Syst.(NSIP)*, **1999** (1999), 368–374.

5. J. Levatic, D. Kocev, M. Ceci, S. Džeroski,Semi-supervised trees for multi-target regression, *Inf. Sci.*, **450** (2018), 109–127.

6. B. Jiang, H. Chen, B. Yuan, X. Yao, Scalable graph-based semi-supervised learning through sparse bayesian model, *IEEE Trans. Knowl. Data Eng.*, **29** (2017), 2758–2771.

7. Z. Zhao, M. Zhao, T. W. S. Chow, Graph based constrained semi-supervised learning framework via label propagation over adaptive neighborhood, *IEEE Trans. Knowl. Data Eng.*, **27** (2013), 2362–2376.

8. J. Xie, B. K. Szymanski, Community detection using a neighborhood strength driven label propagation algorithm, *IEEE Netw. Sci. Work.(NSW)*, West Point, NY, USA, May, **2011** (2011), 188–195.

9. Z. H. Wu, Y. F. Lin, S. Gregory, H. Y. Wan, S. F. Tian, Balanced multi-label propagation for overlapping community detection in social networks, *J. Comput. Sci. Technol.*, **27** (2012), 468–479.

10. S. M. Kim, P. Pantel, L. Duan, S. Gaffney, Improving web page classification by label-propagation over click graphs, *Proc. 18th ACM Conf. Inform. Knowl. Manag.(CIKM)*, Hong Kong, China, November, **2009** (2009), 1077–1086.

11. S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, J. Reynar,Building a sentiment summarizer for local service reviews, (2008).

12. V. Badrinarayanan, F. Galasso, R. Cipolla, Label propagation in video sequences, *IEEE Comput. Soci. Conf. Comput Vis. Pat Rec.(CVPR)*, San Francisco, CA, USA, June, **2010** (2010), 3265–3272.

13. K. Kothapalli, S. V. Pemmaraju, V. Sardeshmukh, On the analysis of a label propagation algorithm for community detection, *Int. Conf. Dist Comput. Netw.(ICDCN)*, Mumbai, Maharastra, India, July, **2013** (2013), 255–269.

14. C. Gong, D. Tao, W. Liu, L. Liu, J. Yang, Label propagation via teaching-to-learn and learning-to-teach, *IEEE Trans. Neural Netw. Learn. Syst.*, **28** (2016), 1452–1465.

15. J. Hao, X. Chen, S. Huang, Y. Jun, Semi-supervised classification algorithm using fuzzy nearest neighborhood label propagation, *Microelectron. Comput.*, **27** (2010), 30–33.

16. X. K. Zhang, C. Song, J. Jia, Z. L. Lu, Q. Zhang, An improved label propagation algorithm based on the similarity matrix using random walk, *Int. J. Mod. Phys. B*, **30** (2016), 1650093.

17. B. Wang, Z. Tu, J. K. Tsotsos, Dynamic label propagation for semi-supervised multi-class multi-label classification, *Proc. IEEE Int. Conf. Comput Vis.(ICCV)*, Berlin, Germany, June, **2013** (2013), 425–432.

18. X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, (2002).
19. Available from: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py.
20. Available from: https://sklearn.apachecn.org/docs/0.21.3/7.html.
21. Available from: https://sklearn.apachecn.org/docs/0.21.3/11.html.
22. Available from: https://sklearn.apachecn.org/docs/0.21.3/10.html.
23. Available from: https://archive.ics.uci.edu/ml/datasets.php.
24. Available from: https://sklearn.apachecn.org/docs/0.21.3/5.html.
25. J. Deovre, Probability and statistics for engineering and science, *Brooks/Cole*, Belmont, CA (1987).

AIMS Press

©2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)