A Study of the Naive Bayes Classification Based on the Laplacian Matrix

Lei Jiang, Peng Yuan, Qiongbing Zhang, Qi Liu

Abstract—Due to Naive Bayes algorithm has good interpretability and performance, it is widely used to deal with classification problems. Naive Bayes assumes that the attributes are independent of each other. But the phenomenon of the correlations between attributes is always exists in fact. In most case the result of the classification will be strong influenced by these correlations. Thus, minimum the correlation among attributes has been deemed as a challenge for the Naive Bayes research community. In this work, we proposed an improved Naive Bayes method which uses the Laplacian matrix to reconstruct the dataset, since the laplacian matrix can describe the spatial relationship between data attributes well. Experiment results are shown that our method can greatly reduce the correlation between attributes.

Index Terms—Naive Bayes, Laplacian matrix, classification, attribute weighting.

I. Introduction

CLASSIFICATION method is an important data analysis technique which is widely used in commodity image recognition, spam short message retrieval, biomedical category judgment, object recognition and data mining[1][2][3][4][5]. As based on the Bayesian theorem, Naive Bayes method has strong mathematical interpretability especially on model building. So Naive Bayes method is very popular to deal with classification problems[6], and it is identified as one of the top ten data mining algorithms [7][8].

Naive Bayes algorithm assumes that the attributes are independent of each other, that is, Naive Bayes method deems that there is no correlation among data features[9]. Obviously, it is not true in real world applications. Furthermore, the correlation between attributes restricts the performance of Naive Bayes algorithm. Thus, an important research area of Naive Bayes algorithm is to reduce the influence of correlation between attributes using different strategies. The main research directions are as follows. The first is feature selection, which selects the best subset of features to reduce the interference of correlation between attributes. And the other is assigning weights to attributes to weaken the correlation. In these ways, there are a large number of Naive Bayes improvement methods have been proposed[10][11][12][13][14][15][16].

We noticed that those proposed Naive Bayes improvement based on feature selection will cut off the relationship between the best feature subset and the non-optimal subset. It ignored the impact of non-optimal subset, thereby limiting the overall classification performance. The mechanism for assigning weights to attributes is that if an attribute is related to multiple other attributes, it may be unilaterally exaggerated its important. The little weight could reduce the importance of this attribute. Then we can get good results. In this way, the problem evolves into how to find the suitable weights.

In graph theory, the Laplacian matrix is a formal expression of the graph structure [17][18], which is always used to discover some of the properties of the graph[19]. Therefore, it is widely applied to dimensionality reduction and clustering[20][21][22][23]. Due to Laplacian matrix is a semi-positive definite matrix with convexity, it has very good properties. When data is mapped from the high-dimensional space to the low-dimensional space by using the Laplacian matrix, the new data set maintains the structure of the original data set. Inspired by this, we consider that the Laplacian matrix can be used to find the structure of attributes in dataset. Then, we can use this structural information to guide the adjustment of the weight of each attribute. Based on this idea, we proposed a new Naive Bayes method, which is called LPNB, to deal with the classification problems. The main contributions of this paper are summarized as follows:

- We improved the performance of Naive Bayes algorithm by introducing a Laplacian matrix to reconstruct the dataset to reduce the correlation between the attributes of samples.
- After we use the Laplacian matrix to represent the structure of sample attributes, we propose a method to optimize the weights of each attribute of samples.
- We conduct experiments on fifteen datasets and compared the results with six competitive methods. The experimental results indicate that the proposed method is superior to the others.

The remaining organization of the paper is as follows. In Section 2, we introduce the related work of Naive Bayes classification method. In Section 3, we propose our improvement on basic Naive Bayes method. In Section 4 we applies our proposed method to classify data, describe the processing steps and analysis the experiment results, and sum up in Section 5.

Manuscript received January 10, 2020; revised September 02, 2020. This work was supported in part by the Ministry of education of Humanities and Social Science project, China (Grant No. 17YJAZH032).

Lei Jiang is with the Key Laboratory of Knowledge Processing and Networked Manufacture, Hunan University of Science and Technology, Xiangtan, 411201, China e-mail: jleihn@hotmail.com.

Peng Yuan and Qi Liu are with the Key Laboratory of Knowledge Processing and Networked Manufacture, Hunan University of Science and Technology.

Qiongbing Zhang is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan

II. Related Work

The Naive Bayes classification method has excellent characteristics, such as simplicity, high efficiency and interpretability, but it is limited by the assumption that the attributes are independent of each other, so that its performance will be greatly affected in complex applications. Most of the current research is to improve performance by reducing the interference of features. There are two main categories: One is to improve the Naive Bayes classification method by selecting the best feature subset; The second is the weight of attribute method, which is to improve the performance of classification results by assigning weights to change the relationship between attributes.

A. improved method based on feature selection

The Naive Bayes method can be improved by feature selection, which can effectively reduce feature size and accelerate the learning process of classifier[24][25][26]. Correlation-based feature selection (CFS) is a commonly used feature selection method[14]. The core ideal of CFS is that a good feature subset contains features highly related to class variables but they are orthogonally related to each other. CFS searches for feature subset space using the best priority strategy and uses the following equation to evaluate the advantages and disadvantages of feature subset s containing k features:

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{1}$$

where $\overline{r_{cf}}$ is the average value of the correlation between features and classes. $\overline{r_{ff}}$ is the average value of the correlation between features and features. $Merit_s$ is a heuristic data to delete irrelevant features to find a feature subset with larger $\overline{r_{cf}}$. Meanwhile, it is also search for a feature subset with smaller $\overline{r_{ff}}$ by removing redundant features.

Then, the feature i's weight W_i is allocated as follows:

$$W_i = \begin{cases} 1, & \text{if feature } A_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

In this way, CFS performs feature selection. When it is applied to Naive Bayes method, Naive Bayes method based on feature selection (CFSNB)[14] is built. Formula (3) is used to calculate classification results.

$$c(x) = arg\max_{c \in C} P(x) \prod_{i=1}^{m} P(a_i|c)^{W_i} \tag{3}$$

B. Improved method based on feature weighting

As feature weighting can weaken the correlation between features, it is often used to improve the classification performance. Then, many methods is proposed to optimal feature weighting to improve the performance of Naive Bayes method[15] [16]. For example, Jiang et al. [27] proposed a very effective deep feature weighting for Naive Bayes method (DFWNB). First, they use the weights obtained from the CFS to the attributes as the common feature weighting methods. Then, they applied these weights to the conditional probability formula to further improve the results:

$$P(a_i|c, W_i) = \frac{\sum_{j=1}^{n} W_i \delta(a_{ji}, a_i)\delta(c_j, c) + 1}{\sum_{j=1}^{n} W_i \delta(c_j, c) + n_i} \tag{4}$$

$$c(x) = arg\max_{c \in C} P(x) \prod_{i=1}^{m} P(a_i|c, W_i)^{W_i} \tag{5}$$

where n_i is the number of features i. As the main idea of DFWNB is feature weighting, it does not use CFS to delete redundant features. Its weight allocation is as follows.

$$W_i = \begin{cases} 2, & \text{if feature } A_i \text{ is selected} \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

In the above two methods, the CFSNB method selects the best characteristic subset and then carries out the Naive Bayes classification. The DFWNB method assigns the weight by selecting the best characteristic subset based on CFSNB method, and applies them into conditional probability formula to improve the Naive Bayes method. They all achieved good results.

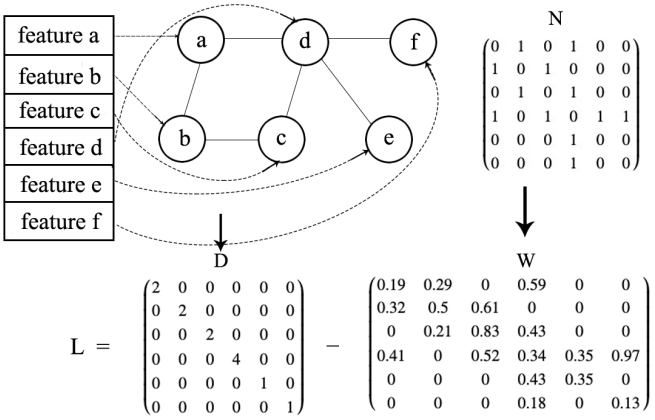

Fig. 1: Laplacian matrix process

C. Laplacian matrix

Laplacian matrix has the property of maintaining the structure of data set in dimensionality reduction. Using this property, many researchers have done a good job. He [21] proposed an algorithm based on laplacian matrix for local feature projection. Dittrich [23] proposed a spectral clustering algorithm based on this characteristic of the Laplacian matrix. He [28] noticed that the two attributes in a dataset have a great correlation if their values in different samples are very close. And then He used the Laplacian matrix to do feature selection. Generally speaking, if the correlation between two feature attributes is higher, the correlation between their corresponding predictions will be higher [29].

In graph theory [17][18], a matrix is often used to represent a graph as it can carry a lot of useful information. Given a graph G with n vertices, its Laplacian matrix is obtained by subtracting the degree matrix from the superposition matrix.

$$W_i = \begin{cases} 1, & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are adjacent} \\ 0, & \text{if } i = j, \text{ or } i \neq j \text{ and } i \text{ is not adjacent j} \end{cases} \tag{7}$$

$$L = D - W \tag{8}$$

Where the adjacency matrix $W = [w_{ij}]$, the degree matrix $D = diag\{d_1, \ldots, d_n\}, d_i = \sum_{j=1} w_{ij}$.

$$x^T Lx = x^T Dx - x^T Wx = \frac{1}{2} \sum_{i,j=1}^{m} w_{ij}(x_i - x_j)^2 \tag{9}$$

When we calculate the distance between two nodes with weights, we can get the equation (9). From it, we can know that the original structure will still be maintained.

III. Approach

In this paper, we propose a Laplacian matrix based Naive Bayes method (LPNB) to deal with classifying problem. This proposal is based on the following considerations. An important assumption of Naive Bayes method is that the attributes are independent of each other. However, in real classification problem, some attributes and even the entire attributes are related, they are not completely independent. Therefore, we need to find a suitable weight allocation method to adjust the attributes to reduce the impact aforementioned. Here we use Laplacian matrix to describe the attributes' structure of the dataset to be classified, and then optimize to find the appropriate weights to reconstruct the dataset. Finally, Naive Bayes is applied for classification.

A. Naive Bayes

Given an instance x, attributes vector$< a_1, a_2, \ldots, a_m >$, and label set C, Naive Bayes method uses the following formula to classify x:

$$c(x) = arg \max_{c \in C} P(x) \prod_{i=1}^{m} P(a_i|c) \tag{10}$$

where $P(c)$ is a prior probability and $P(a_i|c)$ is a conditional probability. Their calculation formulas are as follows:

$$P(c) = \frac{\sum_{j=1}^{n} \delta(c_j, c)}{n} \tag{11}$$

$$P(a_i|c) = \frac{\sum_{j=1}^{n} \delta(a_{ji}, a_i)\delta(c_j, c) + 1}{\sum_{j=1}^{n} \delta(c_j, c) + n_i} \tag{12}$$

where n is the number of training instances, and $\delta(\cdot)$ is a binary function, which takes 1 if its two parameters are identical, otherwise it takes zero.

B. Laplacian matrix process

In Laplacian matrix processing, we first need to construct a Laplacian matrix from the data set which will be classified, and then assign weights to each attribute accordingly. The process is shown in Fig1.

1) construct Laplacian matrix: We construct an adjacency graph G, where each node is an attribute in the dataset. That is, the kNN method is used to find the adjacency attributes of a specified attribute node. We set the k of kNN is refer to the literature [30], i.e. $k = \sqrt{n}$. Here, n is the number of attributes. Then, we can build an adjacency graph G. For detailed methods, please refer to the literature [31].

Furthermore, we get the adjacency matrix N and degree matrix D from G. Then, the Laplacian matrix can be obtained from the following formula.

$$L = D - N \tag{13}$$

2) Get weight matrix: The Naive Bayes method requires attributes to be independent of each other, but it is very common to have attribute correlation in the actual datasets. The Laplacian matrix can represent the attribute relationship well, which means that it is possible to reduce the influence of attribute correlation on the Naive Bayes method by adjusting the data in Laplacian matrix. It is generally believed that the more related the attributes are in the data set, the greater the correlation degree of the adjacency matrix. Therefore, we consider the adjacency matrix N as an expression of the weights between attribute relationships. Then, we adjust N to get the weight matrix W. And L can be obtained by the following formula.

$$L = D - W \tag{14}$$

We obtain W by randomly adjusting the non-zero values in N. Namely, we use $\{a_{ij}|a_{ij} \in N, a_{ij} \neq 0\}$ as the initial position to randomly generate particles, and use the classification accuracy as the target function. Then the particle swarm optimization algorithm (PSO) in literature [32] is used to iterate 1000 times to get the result. Herein, we constrain the search space of a_{ij} to $(-cs, cs)$, for example $(-1, 1)$. Each particle updates its position and velocity according to the following equations (parameter value: $w = 1, c_1 = 0.49, c_2 = 1.49$):

$$\begin{cases} v_{i+1} = wv_i + c_1 r_1(pbest_i - x_i) + c_2 r_2(gbest_i - x_i) \\ x_{i+1} = x_i + v_i \end{cases} \tag{15}$$

Next, we reconstruct the dataset T using the equation 16. Hereafter, the new dataset T' is classified by the Naive Bayes to obtain the final result.

$$T' = TL \tag{16}$$

Herein, the main consideration is that Naive Bayes method requires that attributes be independent of each other. Therefore, we use PSO to optimize the weight of each side on N to obtain the weight matrix W. Ideally, the weight matrix W makes the attributes approximately independent, and also expresses the importance of each attribute in the classification problem. In this way, the classification of the transformed T' will have a better result.

Algorithm 1 Pseudo of LPNB

Input: training samples T, test samples X,
control parameter cs, the number of iteration C
Output: Class labels of X
1. Use kNN method to calculate and generate laplacian matrix L
2. For $i = 1, C$:
3. Control each valued element of the Laplacian matrix, conduct particle swarm search in the search space controlled by the search parameter cs, then obtain the new laplacian matrix L
4. Multiply L and data T to get new data set T'
5. Apply T' to NB, get the prediction result, and feed back to the particle swarm optimization
6. End for
7. Apply the obtained L to NB algorithm of test dataset X

TABLE I: Experimental datasets

Dataset	Abbr.	instances	attributes	categories
abalone	abal	4177	8	3
balance-scale	bala	625	4	3
glass	glass	214	10	7
heart	heart	270	13	2
hepatits	hepa	155	20	2
ionosphere	iono	351	34	2
iris	iris	150	5	3
lymphography	lymp	148	18	4
primary-tumor	prim	339	18	21
segmentation	segm	210	19	7
sonar	sonar	208	60	2
spect	spect	187	23	2
wave	wave	5000	22	3
wdbc	wdbc	569	31	2
wpbc	wpbc	198	34	2

IV. Experiments

In the experiments, we evaluate the performance of our method from two aspects. One is to verify the performance of LPNB from different data dimensions. Secondly, it is compared with the classic classification algorithms, Naive Bayes and its latest improvement method.

A. Experimental data and evaluation criteria

In our experiment, we used 15 datasets which come from UCI machine learning repository. In order to evaluate the performance of the algorithm adequately, the data sets we choose have various types. Their dimensions are covered from low to high and the number of classes ranges is from 2 to 21. And for missing values in these datasets, we replaced it with 0. The detail description is provided in Table I.

For benchmark, we chose the standard Naive Bayes method, the latest Naive Bayes improved methods CFSNB and DFWNB and the classic classification method, such as kNN, SGD and Decision Tree[33][34][35][36]. At the same time, accuracy, recall and f1_score are used to evaluate each method[37]. Their parameters are defined with reference to Table II, and the formulas are as follows.

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{17}$$

TABLE II: The output of classification

	1	0
1	true positive	false positive
0	false positive	true positive

TABLE III: Accuracy of methods with different cs parameters in LPNB

	(-1, 1)	(-2, 2)	(-3, 3)	(-4, 4)	(-5, 5)
abalone	55.28	55.54	55.81	55.78	55.76
balance-scale	91.68	93.12	92.96	92.80	92.64
glass	71.96	71.03	72.90	72.90	72.43
heart	86.30	86.67	87.78	86.30	88.15
hepatits	87.10	87.74	89.03	89.03	89.68
ionosphere	96.30	95.73	94.87	96.87	95.44
iris	99.33	99.33	99.33	99.33	99.33
lymphography	93.24	92.57	93.24	93.92	95.27
primary-tumor	64.01	65.19	65.49	63.72	65.19
segmentation	87.14	89.05	89.52	86.19	88.09
sonar	94.23	93.75	92.31	90.87	92.31
spect	96.26	95.72	95.19	96.26	95.72
wave	86.38	86.76	86.28	86.60	86.52
wdbc	95.96	95.43	95.25	95.25	95.61
wpbc	77.27	77.27	76.77	76.26	76.26

TABLE IV: Recall of methods with different cs parameters in LPNB

	(-1, 1)	(-2, 2)	(-3, 3)	(-4, 4)	(-5, 5)
abalone	55.41	56.11	56.40	56.21	56.26
balance-scale	91.72	92.77	92.65	92.53	92.42
glass	71.80	64.98	69.94	72.40	69.33
heart	85.75	86.42	87.25	85.83	87.83
hepatits	82.62	85.34	85.00	88.47	83.09
ionosphere	96.30	95.10	94.60	96.51	94.70
iris	99.33	99.33	99.33	99.33	99.33
lymphography	96.31	95.90	96.31	96.51	97.43
primary-tumor	73.64	72.58	76.27	75.12	72.76
segmentation	87.14	89.05	89.52	86.19	88.09
sonar	94.14	93.62	92.21	90.79	92.27
spect	91.88	88.55	85.21	91.88	95.72
wave	86.35	86.74	86.25	86.56	86.49
wdbc	95.44	95.44	94.40	94.21	95.61
wpbc	68.25	71.91	70.11	71.98	69.05

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{18}$$

$$f1\_score = 2\frac{recall \times precision}{recall + precision} \tag{19}$$

In order to evaluate the quality of the methods from the m datasets in the experiments, we sort the results obtained by each method in the same dataset. Then accumulate the rankings in different datasets. At this point, the smallest is best. That is, for the dataset T_i, the result of the method j is ranked as k, then recorded as $r_{ij} = k$. In this way, the evaluation score of the method j in the entire datasets can be expressed by the following formula.

$$score_j = \sum_{i=1}^{m} r_{ij} \tag{20}$$

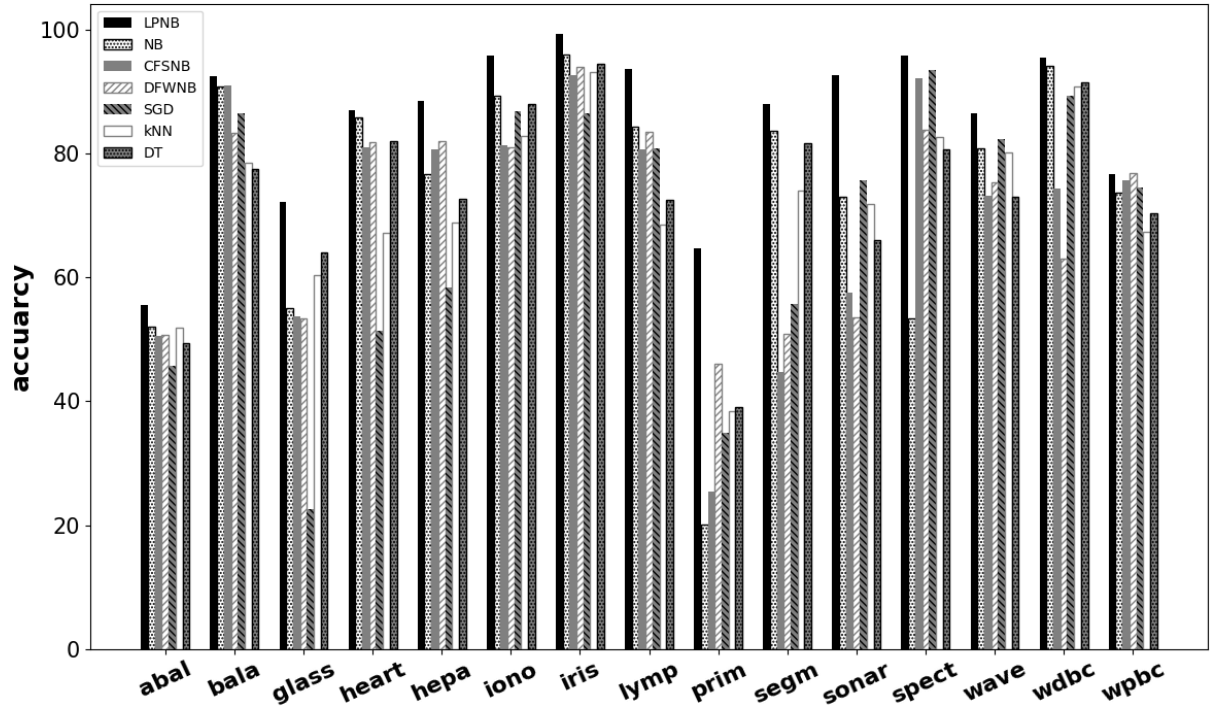

Fig. 2: The accuracy of each method in different data sets

TABLE V: f1_score of methods with different cs parameters in LPNB

	(-1, 1)	(-2, 2)	(-3, 3)	(-4, 4)	(-5, 5)
abalone	53.36	54.06	54.21	54.3	54.21
balance-scale	92.47	93.42	93.33	93.33	93.17
glass	70.61	70.06	71.94	72.96	72.29
heart	86.23	86.65	87.72	86.25	88.14
hepatits	87.37	88.17	89.21	89.50	89.55
ionosphere	96.30	95.71	94.88	96.86	95.42
iris	99.33	99.33	99.33	99.33	99.33
lymphography	93.23	92.54	93.23	93.88	95.27
primary-tumor	63.76	64.09	64.37	62.97	64.42
segmentation	86.30	89.03	89.35	85.11	87.26
sonar	94.23	93.74	92.30	90.86	92.31
spect	96.41	95.84	95.26	96.41	95.95
wave	86.31	86.68	86.23	86.51	86.51
wdbc	95.94	95.40	95.23	95.22	95.58
wpbc	77.19	77.92	77.23	77.17	76.66

B. The selection of parameters *cs*

In our proposed LPNB, we introduce *cs* parameter to control how to maintain the structure of the original dataset as much as possible. The *cs* parameter is used to control the range of weights, that is, the particle swarm algorithm search range for each non-zero value in the adjacency matrix N. In order to investigate the effect of different values of *cs* on the results, *cs* were set as (-1, 1), (-2, 2), (-3, 3), (-4, 4), (-5, 5), and iterated 10,00 times to search for the optimal Laplacian matrix respectively. From the results in Table [III-V], we found that the difference is small when LPNB takes different *cs* ranges. So we make the average of models and used the average values as the final results.

C. The performance of LPNB in different number of dimensions and class labels

1) The performance of different dimensions: In order to evaluate the performance of the LPNB algorithm in different dimensions, we illustrate the experimental results in ascending order of the datasets' dimensions in Fig 5. Obviously, the LPNB performs better than the others on the whole. This shows that LPNB has good performance from low-dimensional to high-dimensional data sets. At the same time, it can be seen that all the algorithms performed poorly on the abalone dataset (8-dimensional) and the primary-tumor dataset (18-dimensional). For the abalone dataset, its last dimension are discrete data, but the others are continuous data. Even worse, the last value of dimensional data is much larger than the others. It can be considered that the current algorithm is relatively poor in processing these data sets where discrete data and continuous data are mixed. It may be an important direction for future research. As for the primary-tumor dataset, it has a lot of incomplete data, and we replace these missing data with 0. In addition, the primary-tumor data set has only three values. Obviously, neither LPNB nor the methods we used for comparison have a good way to deal with this situation, so their results are poorly. For LPNB, the reason is that the Laplacian matrix established by the Euclidean distance cannot maintain the structure of the

TABLE VI: The accuracy of each method in different data sets(acc means accuracy, r means rank)

	LPNB		NB		CFSNB		DFWNB		SGD		kNN		DT	
	acc	r	acc	r	acc	r	acc	r	acc	r	acc	r	acc	r
abalone	**55.63**	1	52.09	2	50.65	5	50.77	4	45.74	7	51.87	3	49.43	6
balance-scale	**92.64**	1	90.88	3	91.13	2	83.39	5	86.54	4	78.53	6	77.56	7
glass	**72.24**	1	55.14	4	53.81	5	53.33	6	22.64	7	60.38	3	64.15	2
heart	**87.04**	1	85.93	2	81.11	5	81.85	4	51.49	7	67.16	6	82.09	3
hepatits	**88.52**	1	76.77	4	80.67	3	82.00	2	58.44	7	68.83	6	72.73	5
ionosphere	**95.84**	1	89.46	2	81.43	6	81.14	7	86.86	4	82.86	5	88.00	3
iris	**99.33**	1	96.00	2	92.67	6	94.00	4	86.49	7	93.24	5	94.59	3
lymphography	**93.65**	1	84.46	2	80.71	5	83.57	3	80.82	4	68.49	7	72.60	6
primary-tumor	**64.72**	1	20.06	7	25.45	6	46.06	2	34.91	5	38.46	4	39.05	3
segmentation	**88.00**	1	83.81	2	44.76	7	50.95	6	55.77	5	74.04	4	81.73	3
sonar	**92.69**	1	73.08	3	57.50	6	53.50	7	75.73	2	71.84	4	66.02	5
spect	**95.83**	1	53.48	7	92.22	3	83.89	4	93.55	2	82.80	5	80.65	6
wave	**86.51**	1	80.92	3	73.16	6	75.42	5	82.47	2	80.23	4	73.07	7
wdbc	**95.50**	1	94.20	2	74.46	6	63.04	7	89.44	5	90.85	4	91.55	3
wpbc	76.77	2	73.74	5	75.78	3	**76.84**	1	74.49	4	67.35	7	70.41	6
score		16		50		64		67		72		73		68

TABLE VII: Recall of comparative method(rc means recall, r means rank)

	LPNB		NB		CFSNB		DFWNB		SGD		kNN		DT	
	recall	r	recall	r	recall	r	recall	r	recall	r	recall	r	recall	r
abalone	**56.08**	1	53.90	2	49.59	5	51.54	3	48.87	7	51.47	4	49.27	6
balance-scale	**92.42**	1	65.74	3	66.03	2	60.25	5	62.34	4	55.45	7	59.06	6
glass	**69.69**	1	61.58	3	53.81	4	38.25	6	28.85	7	48.96	5	64.96	2
heart	**86.62**	1	85.58	2	79.77	5	81.57	4	57.03	7	65.54	6	82.58	3
hepatits	**84.90**	1	79.59	3	80.67	2	73.65	4	43.80	7	50.36	6	71.26	5
ionosphere	**95.36**	1	86.19	2	81.43	5	81.71	6	82.67	4	76.93	7	85.84	3
iris	**99.33**	1	96.00	2	92.67	6	94.08	4	88.27	7	93.77	5	94.68	3
lymphography	**96.49**	1	91.38	2	71.87	4	84.25	3	64.98	5	45.69	6	36.42	7
primary-tumor	**74.07**	1	47.89	2	8.44	7	34.33	3	19.05	6	19.81	5	24.20	4
segmentation	**88.00**	1	83.81	2	47.35	7	49.58	6	55.82	5	76.71	4	81.92	3
sonar	**92.61**	1	74.06	3	57.60	6	53.68	7	75.43	2	71.35	4	66.08	5
spect	90.65	2	74.71	6	**92.22**	1	86.84	3	81.03	5	83.05	4	74.14	7
wave	**86.48**	1	80.84	3	73.06	6	75.42	5	82.24	2	80.30	4	73.04	7
wdbc	**95.02**	1	93.17	2	72.02	6	63.03	7	85.29	5	90.27	4	91.68	3
wpbc	70.26	2	67.39	3	53.48	6	**76.84**	1	57.77	5	48.82	7	60.70	4
score		17		40		72		67		78		78		68

TABLE VIII: f1_score of comparative method(f1 means f1_score, r means rank)

	LPNB		NB		CFSNB		DFWNB		SGD		kNN		DT	
	f1	r	f1	r	f1	r	f1	r	f1	r	f1	r	f1	r
abalone	**54.03**	1	47.73	6	50.68	3	49.76	4	36.61	7	51.92	2	49.53	5
balance-scale	**93.14**	1	65.74	4	87.45	2	83.39	3	62.34	5	55.45	7	59.06	6
glass	**71.57**	1	61.58	3	48.91	6	51.83	4	28.85	7	48.96	5	64.96	2
heart	**87.21**	1	85.58	2	80.54	5	81.96	4	57.03	7	65.54	6	82.58	3
hepatits	**88.76**	1	79.59	3	72.70	4	83.08	2	43.80	7	50.36	6	71.26	5
ionosphere	**95.83**	1	86.19	2	81.79	6	82.98	4	82.67	5	76.93	7	85.84	3
iris	**99.33**	1	96.00	2	92.67	6	93.92	4	88.27	7	93.77	5	94.68	3
lymphography	**93.63**	1	91.38	2	56.86	4	49.98	5	64.98	3	45.69	6	36.42	7
primary-tumor	**63.98**	1	47.89	2	10.89	7	44.73	3	19.05	6	19.81	5	24.20	4
segmentation	**87.41**	1	83.81	2	41.40	7	52.08	6	55.82	5	76.71	4	81.92	3
sonar	**92.69**	1	74.06	3	56.86	6	49.98	7	75.43	2	71.35	4	66.08	5
spect	**95.97**	1	62.90	7	88.60	3	72.50	6	93.97	2	86.67	4	85.01	5
wave	**86.36**	1	79.72	4	71.77	7	74.85	5	82.09	2	80.31	3	73.11	6
wdbc	**95.44**	1	94.16	2	74.12	6	48.98	7	88.93	5	90.86	4	91.61	3
wpbc	**77.10**	1	74.55	2	68.62	5	67.05	6	71.58	3	64.08	7	70.61	4
score		15		46		77		70		73		75		64

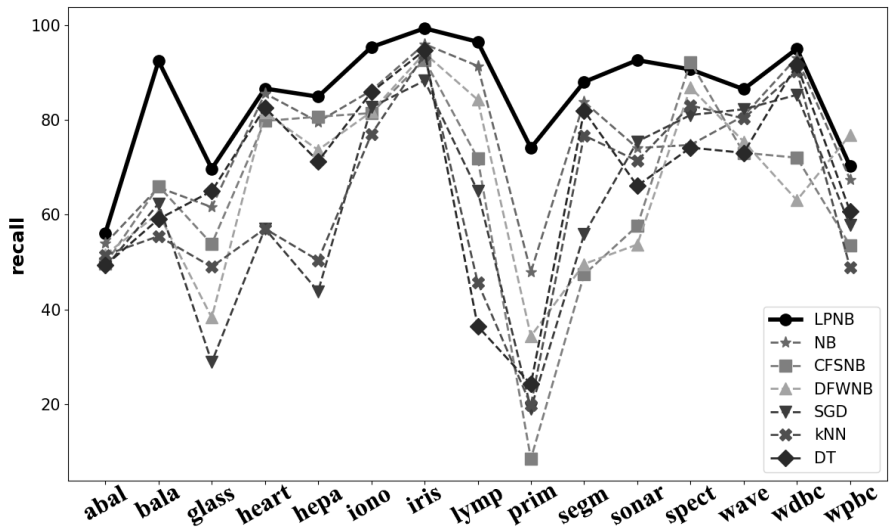

Fig. 3: Recall of different methods

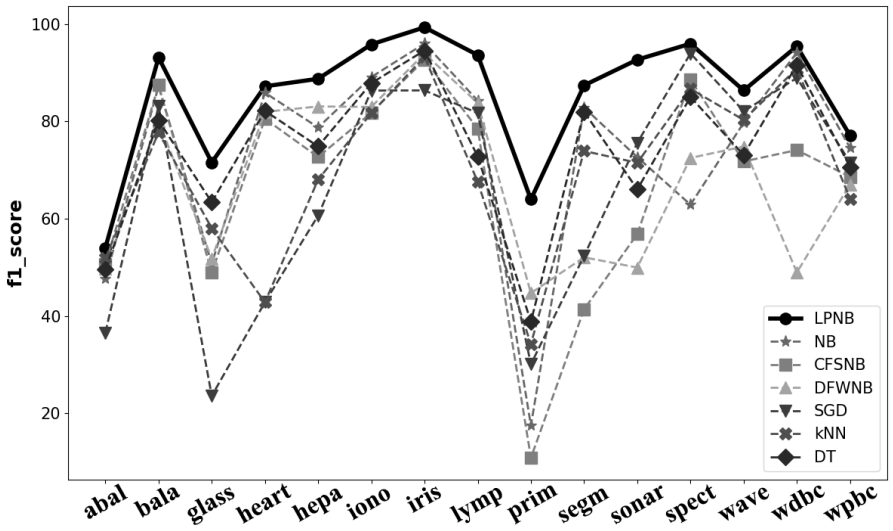

Fig. 4: f1_score of different methods

original dataset well in this case.

2) The performance of different numbers of class label: In this section, we observe the performance of LPNB from the perspective of different numbers of classes. In Fig 6, we show the accuracy of each algorithm according to the number of classes from small to large. It can be seen that LPNB is superior to other methods in the case of multiple classifications, though, the performance in different dimensions is fluctuates. Then, we argue that the LPNB method performs well in different class labels.

D. Competing methods

It can be seen from Table VI-VIII that the accuracy score of LPNB is 16, the recall rate is 17, and the f1_score score is 15. Namely, the rank of LPNB is highest among every evaluation criteria. Therefore, we can conclude that LPNB has the best performance among all the compared algorithms. In addition, we compared and analyzed the performance of the algorithm in three area: 1) comparison with Naive Bayes, 2) the latest improved methods of Naive Bayes CFSNB, DFWNB, and 3) classic classification algorithms kNN, SGD and Decision Tree.

First, from the comparison of LPNB and Naive Bayes, the improvement of LPNB is very obvious. As shown in Table VI, the classification performance of LPNB in all datasets is very good, its classification accuracy exceeds Naive Bayes, and the average classification accuracy of each LPNB dataset is increased by 11.65%. Similarly, the comparison between the two results in the recall rate and f1_score also supports this identification. Therefore, we believe that LPNB implicitly improves the independence between attributes by adjusting the weights of the

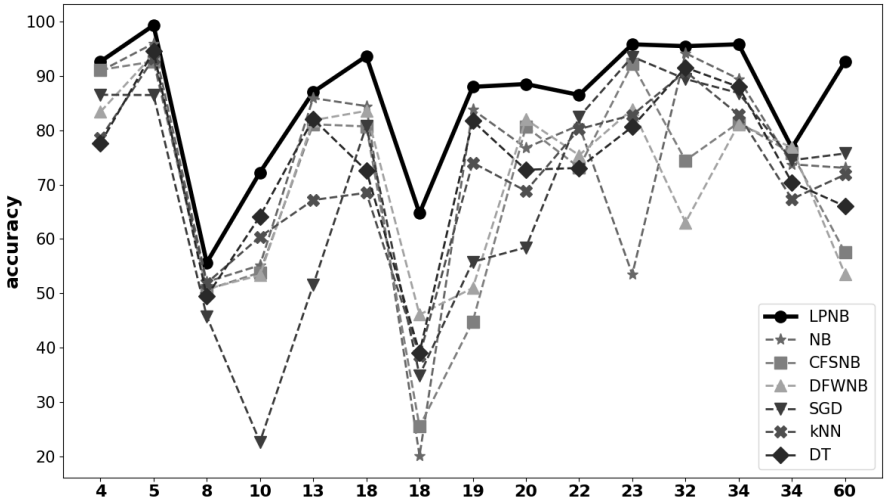

Fig. 5: LPNB comparison of different dimensions in accuracy

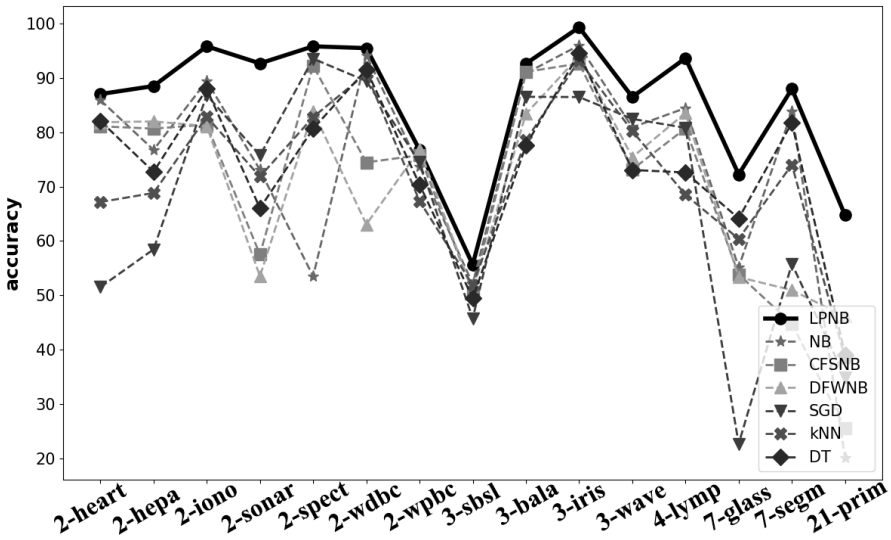

Fig. 6: Different Category cs corresponds to accuracy

attributes.

Secondly, for Naive Bayes's latest improved algorithms, CFSNB and DFWNB, LPNB is still better than them. It can be seen from Table VI that the accuracy of LPNB is 15.29% and 15.01% better than CFSNB and DFWNB methods in average, respectively. For the evaluation of recall and f1_score, LPNB's score is significantly ahead of CFSNB and DFWNB (Table VII-VIII). From Figs. [3-4], LPNB is intuitively superior to the other two Naive Bayes methods. For example, in the wdbc dataset, we have enhanced independence between all attributes, so it is better than the two improved methods of CFSNB and DFWNB. Similarly, it can be seen from Fig 5-6 that under different dimensions and different multi-classification problems, LPNB is better than CFSNB and DFWNB. Because CFSNB selects features, it ignores the impact of unselected attributes on the final classification label. For DFWNB, it finds the optimal feature subset of each class and weights them separately. But, there is no weight distinction for each different attributes, and DFWNB ignores the relationship between the features in the two classes and reduces the performance of the algorithm. So LPNB performs better than CFSNB and DFWNB. Overall, LPNB may only be slightly worse than DFWNB on wpbc in all data sets. The wpbc is a very unbalanced data set (positive example: negative example = 151: 47). The DFWNB algorithm is shown the property that it will be attracted by big classes on the wpbc data set, so it is better than LPNB in the recall rate, but it is worse than LPNB in the f1_score evaluation. This shows that our method (LPNB) has better performance

on unbalanced data sets, especially in those cases where a small class require higher accuracy. In the rest of the datasets, the classification results show that LPNB has good performance.

Finally, we compared LPNB with other classic classification algorithms such as kNN, SGD, Decision Tree to observe its performance. It can be seen from Table VI-VIII and Fig 2-4 that LPNB has obvious advantages over kNN, SGD, Decision Tree in the performance of accuracy, recall, and f1_score. On the average accuracy of 15 data sets, LPNB keeps ahead SGD 17.30%, is exceeding kNN 13.87% and is better than Decision Tree 12.09%. The number of iterations will affect the parameter optimization of the SGD algorithm, and its parameters are sensitive. Minor perturbations of the parameters can cause large changes in its results. Therefore, SGD needs to add regular terms to improve the results. However, there are no standards for determining the number of iterations and selecting an appropriate regular term. For SGD in this paper, we refer to the regular terms and iterations commonly used by most researchers. They may not be the best match for our dataset. Therefore, the results of SGD are worse than the LPNB. The results of f1_score and recall show that the LPNB has better classification stability than SGD. Since kNN is unsupervised learning method and it is affected by the k value. KNN is a distance-based classification method. However, Euclidean distance which used in this paper for kNN is difficult to describe the data characteristics of different data sets. Furthermore, the k value of kNN is difficult to set accurately, which also affects the effectiveness of the algorithm. That is why kNN is inferior to LPNB. For the Decision Tree method, the results of LPNB are better than it from the three evaluation indicators. We carefully analyze the reason that the Decision Tree method also selects a subset of features for classification. Although those discarded attributes do not support the classification results obviously, they are still effective supplements for classification. Therefore, when the LPNB enhances the independence between attributes through the laplacian matrix, the LPNB using full attributes for classification is better than the Decision Tree.

Overall, the experiments shows that LPNB performs well in high-dimensional, multi-class situations. LPNB also performs well compared with Naive Bayes method and its recent excellent improvement approaches. It also has advantages compared among the classic SGD, kNN and Decision Tree methods. At the same time, it can be seen from the experiment that there are two difficulties in the current classification problem. Similar to other existing classification method, if the dataset mixed the discrete data with continuous data, or if there are large orders of magnitude differences between data in the same data set, LPNB does not perform well, but it is better than other existing methods.

V. Conclusion

As a classic algorithm, Bayesian classification is an important research ad-hoc in this area. This work proposes a new method to enhance the independence between attributes in a dataset by using a Laplacian matrix to improve Naive Bayes algorithm. The proposed method can make Bayesian method running well: the attributes are independent each other. We conducted experiments on 15 datasets and compared the proposed method with six competitive methods. The experimental results are shown that the proposed method outperforms other alternatives in most cases in terms of effectiveness.

References

[1] A. Wood, V. Shpilrain, K. Najarian, and D. Kahrobaei, "Private naive bayes classification of personal biomedical data: Application in cancer data analysis," Computers in biology and medicine, vol. 105, pp. 144–150, 2019.

[2] J. Chen, Z. Dai, J. Duan, H. Matzinger, and I. Popescu, "Naive bayes with correlation factor for text classification problem," arXiv preprint arXiv:1905.06115, 2019.

[3] C.-z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving naive bayes classifiers secure against the substitution-then-comparison attack," Information Sciences, vol. 444, pp. 72–88, 2018.

[4] B. Ning, W. Junwei, and H. Feng, "Spam message classification based on the naïve bayes classification algorithm," IAENG International Journal of Computer Science, vol. 46, no. 1, pp. 46–53, 2019.

[5] A. Attarian, G. Danis, J. Gronsbell, G. Iervolino, L. Layne, D. Padgett, and H. Tran, "Baseball pitch classification: a bayesian method and dimension reduction investigation," IAENG Transactions on Engineering Sciences, pp. 392–399, 2014.

[6] S. Xu, "Bayesian naïve bayes classifiers to text classification," Journal of Information Science, vol. 44, no. 1, pp. 48–59, 2018.

[7] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, "Naive bayes classification of uncertain data," in 2009 Ninth IEEE International Conference on Data Mining. IEEE, 2009, pp. 944–949.

[8] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," Procedia Computer Science, vol. 60, pp. 708–713, 2015.

[9] I. Rish et al., "An empirical study of the naive bayes classifier," in IJCAI 2001 workshop on empirical methods in artificial intelligence, vol. 3, no. 22, 2001, pp. 41–46.

[10] D. Dash and G. F. Cooper, "Model averaging for prediction with discrete bayesian networks," Journal of Machine Learning Research, vol. 5, no. Sep, pp. 1177–1203, 2004.

[11] L. Jiang, Z. Cai, D. Wang, and H. Zhang, "Improving tree augmented naive bayes for class probability estimation," Knowledge-Based Systems, vol. 26, pp. 239–245, 2012.

[12] A. Askari, A. d'Aspremont, and L. E. Ghaoui, "Naive feature selection: Sparsity in naive bayes," arXiv preprint arXiv:1905.09884, 2019.

[13] J. Bai, Y. Yang, and J. Xie, "Improved naïve possibilistic classifiers for imprecise data," IAENG International Journal of Computer Science, vol. 45, no. 1, pp. 153–163, 2018.

[14] L. Jiang, L. Zhang, C. Li, and J. Wu, "A correlation-based feature weighting filter for naive bayes," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 2, pp. 201–213, 2018.

[15] C.-H. Lee, F. Gutierrez, and D. Dou, "Calculating feature weights in naive bayes with kullback-leibler measure," in 2011 IEEE 11th International Conference on data mining. IEEE, 2011, pp. 1146–1151.

[16] L. Chen and S. Wang, "Automated feature weighting in naive bayes for high-dimensional data classification," in Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012, pp. 1243–1252.

[17] G. J. Mahdi, A. Chakraborty, M. E. Arnold, and A. G. Rebelo, "Efficient bayesian modeling of large lattice data using spectral properties of laplacian matrix," Spatial statistics, vol. 29, pp. 329–350, 2019.

[18] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The laplacian spectrum of graphs," Graph theory, combinatorics, and applications, vol. 2, no. 871-898, p. 12, 1991.

[19] D. B. West et al., Introduction to graph theory. Prentice hall Upper Saddle River, NJ, 1996, vol. 2.

[20] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," Neural computation, vol. 15, no. 6, pp. 1373–1396, 2003.

[21] X. He, C. Zhang, L. Zhang, and X. Li, "A-optimal projection for image representation," IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 5, pp. 1009–1015, 2015.
[22] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in Advances in neural information processing systems, 2002, pp. 585–591.
[23] T. Dittrich, P. Berger, and G. Matz, "Semi-supervised spectral clustering using the signed laplacian," in 2018 52nd Asilomar Conference on Signals, Systems, and Computers. IEEE, 2018, pp. 1413–1417.
[24] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and naive bayes," in ICML, vol. 99, 1999, pp. 258–267.
[25] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in Icml, vol. 1, 2001, pp. 74–81.
[26] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive bayes for text categorization," IEEE transactions on knowledge and data engineering, vol. 28, no. 9, pp. 2508–2521, 2016.
[27] L. Jiang, C. Li, S. Wang, and L. Zhang, "Deep feature weighting for naive bayes and its application to text classification," Engineering Applications of Artificial Intelligence, vol. 52, pp. 26–39, 2016.
[28] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in Advances in neural information processing systems, 2006, pp. 507–514.
[29] X. Zhu, H.-I. Suk, and D. Shen, "Matrix-similarity based loss function and feature selection for alzheimer's disease diagnosis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3089–3096.
[30] U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," Water Resources Research, vol. 32, no. 3, pp. 679–693, 1996.
[31] X. He and P. Niyogi, "Locality preserving projections," in Proceedings of the 16th International Conference on Neural Information Processing Systems. MIT Press, 2003, pp. 153–160.
[32] Y. Jiang, T. Hu, C. Huang, and X. Wu, "An improved particle swarm optimization algorithm," Applied Mathematics and Computation, vol. 193, no. 1, pp. 231–239, 2007.
[33] L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, no. 2, p. 1883, 2009.
[34] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.
[35] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE transactions on systems, man, and cybernetics, vol. 21, no. 3, pp. 660–674, 1991.
[36] C. Wang and Y. Yang, "Nearest neighbor with double neighborhoods algorithm for imbalanced classification." International Journal of Applied Mathematics, vol. 50, no. 1, 2020.
[37] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

Lei Jiang received his B. Sc. and M.S. degrees in Computer Science from the Southwest Petroleum University in 1996 and 2005, respectively, and Ph. D. degree from the State Key Laboratory of Software Engineering (SKLSE), Wuhan University, Wuhan, China, in 2012. He is currently an associate professor of Department of Computer Science and Engineering, Hunan University of Science and Technology in Xiangtan, China. His research interests include social computing, evolutionary computation, machine learning and decision support.

Peng Yuan received the BSc degree in computer science and technology from the Hunan University of Science and Technology, China, in 2018. He is currently a member of the Key Laboratory of Knowledge Processing and Networked Manufacturing at the Hunan University of Science and Technology. His research interests include social computing, machine learning and data mining.

Qiongbing Zhang received the B.S. degree from Central South University, Changsha, China, in 2009, and Ph.D. degree from the State Key Laboratory of Software Engineering (SKLSE), Wuhan University, Wuhan, China, in 2017. He is currently a lecturer at the School of Computer Science and Technology, Hunan University of Science and Technology. His research interests include intelligent computing, communication routing and machine learning.

Qi Liu received the BSc degree in computer science and technology from the Hunan University of Humanities,Science and Technology, China, in 2016. She is currently a member of the Key Laboratory of Knowledge Processing and Networked Manufacturing at the Hunan University of Science and Technology. Her research interests include social computing, machine learning and data mining.