2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems

Neighborhood graph embedding for nodes clustering of social network

Shuliang Xu[1], Shenglan Liu[2], Lin Feng[2,*]

1. Faculty of Electronic Information and Electrical Engineering Dalian University of Technology Dalian, China
email: xushulianghao@126.com

2. School of Innovation and Entrepreneurship Dalian University of Technology Dalian, China
email: fenglin@dlut.edu.cn

***Abstract*—Graph embedding is an important dimension reduction method for high-dimensional data. In this paper, a neighborhood graph embedding algorithm is proposed and it is applied in data clustering. Different from the traditional graph embedding algorithms, a dependence degree of node is defined and it represents the dependence of two nodes; the adjacency matrix of graph is determined by dependence degree. Then a new graph embedding is proposed. After transformation matrix is solved, the weight of each attribute can also be determined from transformation matrix. Finally, the data is partitioned into clusters by clustering algorithm with weighting distance. The proposed algorithm and comparison algorithms are executed on the real social network data sets. The experimental results show that the proposed algorithm outperformances the comparison algorithms and it proves that the proposed algorithm is effective for data clustering in social network.**

***Index Terms*—Neighborhood, social network, clustering analysis, graph embedding, dimension reduction**

I. Introduction

In information society, social network is an important resource and the relationships of peoples in different places are contained in the network [1]–[3]. Social network contains abundant information, we can recommend merchandises to users and find friends with common interests by analyzing network which means huge commercial value [4]–[6]. However, because of high dimension and high complexity, it makes that it is not easy to mine knowledge and patterns from social network, therefore how to efficiently analyze social network has attracted much attention from researchers and many related works have been published [7]–[9].

Boden et al. propose a graph clustering algorithm which can track the graph evolvement called as DB-CSC [10]; DB-CSC algorithm is a development of DBCAN algorithm [11] and uses the density to generate clusters; it introduces the node attribute information to improve the performance; the change of clusters can be detected by hypothesis test. Qian et al. propose a significance degree for nodes in complex network which keep global connectivity [12]; complex network is presented as an adjacency matrix, then the low rank presentation of adjacency matrix can be obtained by non-negative matrix factorization [13]; the significance degree of an edge can be measured by link entropy and an large link entropy means the nodes of an edge have a large possibility in the different communities. Lu et al. propose a community discovery algorithm based on conductivity and modularity for weighted network [14]; the algorithm searches two nodes with maximum weight and generates an initial community; then the neighbors of the nodes in the community with maximum weights are potentially partitioned into the community; if conductivity becomes less than the previous value, the neighbors should partitioned into the community; otherwise, the neighbors are seen as a new community. Liu et al. proposed a community detection algorithm based on multiobjective evolutionary algorithm for signed social networks called as MEAs-SN [15]; MEAs-SN uses multiobjective optimization and unsupervised learning methods to discover community structure; if the signed value becomes larger after a node is partitioned into the community, the node is seen as a member of the community; in order to avoid the number of communities too large, the similar communities where the rate of the common nodes is larger than 50% are merged. Whang et al. propose a neighborhood-inflated seed expansion algorithm for overlapping community detection [16]; the algorithm is based on hierarchical clustering algorithm and introduce PageRank [17] to discover community; the double connection core area is determined at first, then the seed points with good conductivity are selected; the community can be discovered by random walk strategy and PageRank algorithm. Liu et al. propose a edge label propagation algorithm in complex network called as ELPA [18], ELPA algorithm is a development of LPA algorithm [19] which can cannot discover overlapping communities; ELPA algorithm introduces the mechanism of edge label propagation to discover overlapping communities and the label of a node can also be determined by the labels of edges. Eustace et al. propose a local neighborhood community detection algorithm in complex networks [20]; for each node, the algorithm finds the local community and all neighborhood nodes which satisfy the α-close relation are added into the community of the node; then the related local communities are merged; finally, the local communities which satisfy the β-close relation are also merged and the result can generate non-overlapping communities. Li et al. proposed a local spectral clustering algorithm [21] to discover the small community in large networks called as LEMON [22]; LEMON incorporates random walk and

*Corresponding author.

978-1-7281-2058-4/19/$31.00 ©2019 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2019.00107

spectral clustering and can discover the small community in a local network which means that it can not scan the full large network. Gleich at al. propose a new local community discovery method [23]; it defines clustering coefficient which is defined according to wedge and uses the personlizaeed page rank algorithm to the result of community discovery; the authors prove that a good community should have a high tail distribution and large clustering coefficient. Liu et al. propose an communities detection algorithm based on network topology [24]; the algorithm is based on LPA algorithm and finds the strong associated communities whose the strength of internal correlation is greater than the strength of external correlation and weak associated communities whose the strength of internal correlation is less than or equal to the strength of external correlation to obtain the initial nodes; for each node, the label is determined by the labels of neighborhood nodes; finally, the labels of all nodes are adjusted by LPA algorithm.

The above works have promoted the development of community discovery. However, the most of the above works are based on the edges of nodes to discover community and does not uses whole neighborhood set of nodes. In this paper, a neighborhood graph embedding for nodes clustering of social network (NGEC) is proposed which transforms community discovery into a clustering problem. In order to reduce the impact of using only edges on the clustering result, a neighborhood graph embedding is introduced to improve the performance of the algorithm. The contributions of this paper are as follows:

- A dependence degree of node is defined which means the significance of a node for anther node and the adjacency matrix of social network is determined by dependence degree; it makes the adjacency matrix contains more information of nodes.
- A new graph embedding algorithm is proposed to decrease the dimension of adjacency matrix and the weight of each attribute can be determined from transformation matrix. The weighted distance is used in the clustering algorithm.
- The proposed algorithm and comparison algorithm are tested on the real social network data sets and the test results show that the proposed algorithm is an effective algorithm for nodes clustering of social network.

The rest of this paper is organized as follows: Section 2 gives a brief introduction about background knowledge; Section 3 explains the detail principles and steps of the proposed algorithm; in Section 4, the proposed algorithm and comparison algorithm are tested on the real social network data sets and the results are analysized; Section 5 concludes the paper and gives some directions of future works.

II. Background knowledge

In this section, LPP algorithm are reviewed [25]. LPP is an important manifold learning algorithm for dimension reduction and it thinks that the samples can be reconstructed according their neighbors in low dimensional embedded space. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\} \in \mathbb{R}^{m\times n}$ be data points and $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n\} \in \mathbb{R}^{d\times n}$ be the projection of $\boldsymbol{X}$ in low dimensional embedded space. For $\forall \boldsymbol{x}_i \in \boldsymbol{X}$, $\boldsymbol{y}_i = \boldsymbol{P}^T\boldsymbol{x}_i$ and $\boldsymbol{P} \in \mathbb{R}^{m\times d}$. After $\boldsymbol{X}$ is projected into the embedded space, the distances of data points should be consistent with the distances in the original space. The problem can be described as follows:

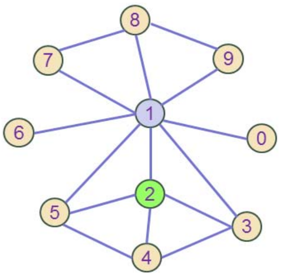

Fig. 1. The structure of a network

$$\min_{\boldsymbol{Y}} \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 \tag{1}$$

where W_{ij} is the weight between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Let $\boldsymbol{D}_{ii} = \sum_{j=1}^{n} W_{ij}$ and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ which is also called as laplacian matrix. The Eq.(1) can express as follow:

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\left\|\boldsymbol{P}^T\boldsymbol{x}_i - \boldsymbol{P}^T\boldsymbol{x}_j\right\|_2^2 \\ &= \sum_{i=1}^{n} \boldsymbol{P}^T\boldsymbol{x}_i\boldsymbol{D}_{ii}\boldsymbol{x}_i^T\boldsymbol{P} - \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{P}^T\boldsymbol{x}_i W_{ij}\boldsymbol{x}_i^T\boldsymbol{P} \\ &= \boldsymbol{tr}\left(\boldsymbol{P}^T\boldsymbol{X}\left(\boldsymbol{D}-\boldsymbol{W}\right)\boldsymbol{X}^T\boldsymbol{P}\right) = \boldsymbol{tr}\left(\boldsymbol{P}^T\boldsymbol{XLX}^T\boldsymbol{P}\right) \end{aligned} \tag{2}$$

Therefore the optimization problem is as follow:

$$\min_{\boldsymbol{P}} \boldsymbol{tr}\left(\boldsymbol{P}^T\boldsymbol{XLX}^T\boldsymbol{P}\right) \quad s.t.\boldsymbol{P}^T\boldsymbol{XDX}^T\boldsymbol{P} = \boldsymbol{1} \tag{3}$$

By lagrange multiplier approach, Eq.(3) can be expressed as the generalized eigenvalue problem:

$$\boldsymbol{X}^T\boldsymbol{LX}^T\boldsymbol{P} = \lambda\boldsymbol{XDX}^T\boldsymbol{P} \tag{4}$$

Therefore $\boldsymbol{P}$ is made up of the d eigenvectors corresponding to d smallest non-zero eigenvalues in Eq.(4).

III. Neighborhood graph embedding for nodes clustering

In this section, we explains the details of NGEC algorithm. At first, we describes the method which determines the weights of adjacency matrix; then we presents the detail principles and steps of the proposed algorithm.

A. Neighborhood dependence of neighborhood node

In social network, the significance of each node is different for different nodes. For example, from Fig.(1), node 1 is a core node because it connects with many nodes which means that node 1 has a large influence on other nodes, therefore it is obvious that node 1 has a larger influence on node 2 than other nodes. In order to state the influence, we has the following definitions. Let $G = (V, E)$ be a graph where V is the set of vertexes and E is the set of edges. $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of G and $A_{ij} = 1$ if $v_i \in V, v_j \in V$ and $e_{ij} \in E$, otherwise, $A_{ij} = 0$.

Definition 1. *(1-order neighborhood) For a graph $G = (V, E)$, $\boldsymbol{A}$ is the the adjacency matrix of* G. *For $\forall v_i \in V$, the 1-order neighborhood of v_i is defined as*

$$\boldsymbol{\Gamma}_V^{(1)}(v_i) = \{v_j | \forall v_j \in V, A_{ij} = 1\} \tag{5}$$

It is obvious that $\boldsymbol{\Gamma}_V^{(1)}(v_i)$ is the set of vertexes which connect with v_i in graph G.

Definition 2. *(1-order neighborhood dependence degree) Let $\forall v_i \in V$, the 1-order neighborhood dependence degree of v_i is defined as*

$$f_V^{(1)}(v_i) = \frac{1}{1 + \exp(-\left|\boldsymbol{\Gamma}_V^{(1)}(v_i)\right|)} \tag{6}$$

where $\left|\boldsymbol{\Gamma}_V^{(1)}(v_i)\right|$ is the cardinality of $\boldsymbol{\Gamma}_V^{(1)}(v_i)$.

Definition 3. *(2-order neighborhood) Let $\forall v_i \in V$, the 2-order neighborhood of v_i in* G *is defined as*

$$\boldsymbol{\Gamma}_V^{(2)}(v_i) = \left\{v_j | v_j \in V, v_j \in \boldsymbol{\Gamma}_V^{(1)}(\boldsymbol{\Gamma}_V^{(1)}(v_i))\right\} \tag{7}$$

Definition 4. *(2-order neighborhood dependence degree) Let $\forall v_i \in V$, the neighborhood dependence degree of v_i is defined as*

$$f_V^{(2)}(v_i) = \frac{1}{1 + \exp(-\left|\boldsymbol{\Gamma}_V^{(2)}(v_i)\right|)} \tag{8}$$

In Eq.(8), $\boldsymbol{\Gamma}_V^{(2)}(v_i)$ is the neighborhood of the neighborhood of v_i.

Definition 5. *The neighborhood dependence degree of v_i is defined as*

$$f_V(v_i) = \frac{1}{1 + \exp\left(-\left(f_V^{(1)}(v_i) + f_V^{(2)}(v_i)\right)\right)} \tag{9}$$

For a graph $G = (V, E)$ and $\forall v_i, v_j \in V$, if v_j has a large impact on v_i, the neighborhood of v_j should have a large effect on $f_V(v_i)$. When v_j is removed from the graph G, the before and after change of $f_V(v_i)$ can be used to measure the significance of v_j on v_i.

Definition 6. *For a graph $G = (V, E)$ and $\forall v_i, v_j \in V$, the dependence degree of v_i on v_j is defined as*

$$\mu_{v_i}(v_j) = \begin{cases} \dfrac{1}{1 + \exp(-\Delta f_V(v_i))} & (v_i \neq v_j) \\ 1 & (v_i = v_j) \end{cases} \tag{10}$$

where $\Delta f_V(v_i) = f_V(v_i) - f_{V - \{v_j\}}(v_i)$.

From Eq.(10), it is known that $\mu_{v_i}(v_j) \in [0, 1]$ and it is the dependence degree of v_i on v_j. A large value of $\mu_{v_i}(v_j) \in [0, 1]$ means a large possibility that v_i and v_j are in the same community and $\mu_{v_i}(v_j) \in [0, 1]$ can be seen the degree of membership to different nodes. Therefore the degree of membership of v_i is as

$$A(v_i) = \frac{\mu_{v_i}(v_1)}{v_1} + \frac{\mu_{v_i}(v_2)}{v_2} + \cdots + \frac{\mu_{v_i}(v_n)}{v_n} \tag{11}$$

For $\forall v_i, v_j \in V$, the weight of the edge e_{ij} can be as $X_{ij} = \mu_{v_i}(v_j)$. Therefore the adjacency matrix $\boldsymbol{A}$ can be changed into a new adjacency matrix $\boldsymbol{X}$.

$$\boldsymbol{X} = \begin{bmatrix} \mu_{v_1}(v_1) & \cdots & \mu_{v_1}(v_n) \\ \vdots & \ddots & \vdots \\ \mu_{v_n}(v_1) & \cdots & \mu_{v_n}(v_n) \end{bmatrix}_{n \times n} \tag{12}$$

The matrix $\boldsymbol{X}$ can be seen as the adjacency matrix of graph G with weights. Because $\boldsymbol{X}$ is a $n \times n$ matrix, for each row of $\boldsymbol{X}$ $(i = 1, 2, \cdots, n)$ is seen as a feature of v_i.

B. Neighborhood graph embedding based on the matrix of neighborhood dependence degree

Let $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n] \in \mathbb{R}^{m \times n}$ and $\boldsymbol{X}_i \in \mathbb{R}^{m \times 1}$. If graph $G = (V, E)$ is a large network, the dimension of $\boldsymbol{X}$ is too high and it is difficult to use $\boldsymbol{X}$ to complete clustering task. Therefore there is a need to decrease the dimension of $\boldsymbol{X}$. Let $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_n]$ is the low-dimension embedding of $\boldsymbol{X}$, $\boldsymbol{Y} \in \mathbb{R}^{d \times n}$ and $\boldsymbol{Y}_i = \boldsymbol{P}^T \boldsymbol{X}_i$ where $\boldsymbol{P} \in \mathbb{R}^{m \times d}$ is a transformation matrix. Therefore the objective optimization function can be expressed as follow:

$$\begin{aligned} &\min_{\boldsymbol{P}} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \left\|\boldsymbol{Y}_i - \boldsymbol{Y}_j\right\|_2^2 + \|\boldsymbol{P}\|_F^2 - \boldsymbol{tr}\left(\boldsymbol{Y}\boldsymbol{Y}^T\right) \\ &s.t.\ \boldsymbol{P}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{P} = \boldsymbol{1} \end{aligned} \tag{13}$$

In Eq.(13), the first term is the error of the projected data, the second term is the generalization ability and the third term is the discernibility. For $\boldsymbol{P} \in \mathbb{R}^{m \times d}$, P_{ij} can be seen as the contribution of the ith attribute of original data for the jth attribute of the projected data. From Eq.(3) and $\boldsymbol{Y} = \boldsymbol{P}^T \boldsymbol{X}$, Eq.(13) can be expressed as

$$\begin{aligned} &\min_{\boldsymbol{P}} \frac{1}{2}\boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{P}\right) + \frac{1}{2}\boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{P}\right) - \frac{1}{2}\boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{P}\right) \\ &s.t.\ \boldsymbol{P}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{P} = \boldsymbol{1} \end{aligned} \tag{14}$$

Therefore the following lagrange function can be defined:

$$\begin{aligned} L = &\frac{1}{2} \cdot \boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{P}\right) + \frac{1}{2} \cdot \boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{P}\right) - \frac{1}{2} \cdot \boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{P}\right) \\ &- \frac{\lambda}{2} \cdot \boldsymbol{tr}\left(\boldsymbol{P}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{P} - \boldsymbol{1}\right) \end{aligned} \tag{15}$$

The partial derivative of L with respect to $\boldsymbol{P}$ is as

$$\frac{\partial L}{\partial \boldsymbol{P}} = \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{P} + \boldsymbol{P} - \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{P} - \lambda \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{P} = 0 \tag{16}$$

From Eq.(16), it has

$$\left(\boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T + \boldsymbol{I} - \boldsymbol{X} \boldsymbol{X}^T\right) \boldsymbol{P} = \lambda \cdot \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{P} \tag{17}$$

From Eq.(17), it is obvious that the matrix $\boldsymbol{P}$ can be solved by the generalized eigenvalues of Eq.(17). The matrix $\boldsymbol{P}$ is made up of the eigenvectors of the *d* smallest non-zero eigenvalues. Therefore the final result is $\boldsymbol{Y} = \boldsymbol{P}^T\boldsymbol{X}$. If $d \leq m$, it can generate a low dimensional data and $\boldsymbol{X}$ is projected on to a low dimensional embedding space.

C. Nodes clustering for social network

After $\boldsymbol{Y}$ is obtained, we can use $\boldsymbol{Y} = \left[\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \cdots, \boldsymbol{y}_n^T\right]_{d\times n}$ as input data to generate clusters. In Eq.(13), $\boldsymbol{P} \in \mathbb{R}^{m\times d}$, therefore P_{ij} can be seen as the projected weight between *i*th original feature and *j*th new feature. The attributes weights $\boldsymbol{\varpi} \in \mathbb{R}^{1\times d}$ of $\boldsymbol{Y}$ are determined as

$$\boldsymbol{\varpi}^T = \frac{1}{m} \cdot \boldsymbol{P}^T \mathbf{1}_{m\times 1} \tag{18}$$

Then $\boldsymbol{\varpi}$ is normalized in $[0, 1]$ and $\forall \varpi_{ij} \in [0, 1]$. Let *num* be the number of clusters, $\boldsymbol{U}_{n\times num} = [u_{ij}]_{n\times num}$ be the membership matrix and $u_{ij} \in [0, 1]$. The objective optimization function of NGEC is as

$$\min_{\boldsymbol{U}} \sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{l=1}^{num} u_{il}\varpi_j \left(y_{ij} - c_{lj}\right)^2 + \sum_{i=1}^{n}\sum_{l=1}^{num} \rho \cdot u_{il} \log u_{il} \tag{19}$$

where $\boldsymbol{C} = \left[\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_{num}\right]^T \in \mathbb{R}^{num\times d}$ and ρ is a parameter and $\rho > 0$ which balances the effect of the second term. In Eq.(19), the first term is the error of clustering result and the second term is the fuzzy entropy of membership. The result of NGEC algorithm is to obtain a result with minimum error and minimum fuzzy entropy.

If $\boldsymbol{U}$ is fixed, c_{lj} is updated as

$$c_{lj} = \sum_{i=1}^{n}\sum_{j=1}^{d} \mu_{il} y_{ij} / \sum_{i=1}^{n} u_{il} \tag{20}$$

If $\boldsymbol{C}$ is fixed, u_{il} is updated as

$$u_{il} = \exp\left(-\frac{\sum\limits_{j=1}^{d} \varpi_j \left(y_{ij} - c_{lj}\right)^2 + \rho}{\rho}\right) \tag{21}$$

Therefore the clustering steps of NGEC algorithm can be summarized as **Algorithm 1**.

Algorithm 1 The clustering steps of NGEC algorithm

```
Input: The adjacency matrix of a graph $\boldsymbol{Y}$; the parameters $\rho$;
    the number of clusters num.
Output: The membership matrix $\boldsymbol{U}$.
Initialize $\boldsymbol{U}$ and $\boldsymbol{C}$;
while The objective function does not converge do
    $\boldsymbol{U}$ is fixed and update $\boldsymbol{C}$ according to Eq.(20) by parallel computing;
    $\boldsymbol{C}$ is fixed and update $\boldsymbol{U}$ according to Eq.(21) by parallel computing;
end while
```

In **Algorithm 1**, $\boldsymbol{U}$ is the final result. A crisp clustering result can be obtained by maximum membership principle and the data points are partitioned into the clusters with maximum membership. In addition, parallel computing is introduced into NGEC algorithm and we use multiprocess method to conduct parallel computing which decreases the time cost of the algorithm. It is obvious that the clustering algorithm of NGEC becomes a k-means fuzzy clustering algorithm with weighted distance if $\rho = 0$.

TABLE I
THE DETAILS OF SOCIAL NETWORK DATA SETS

Data set	Nodes	Edges	Communities
adjnoun	112	425	2
cumulative	332	1,782	Unlabeled
Les Miserables	77	254	Unlabeled
Neural network	297	2,345	Unlabeled
dolphins	62	159	Unlabeled
politics	105	441	3

Theorem 1. ***Algorithm 1** can converge within finite iterations.*

Proof. Let $\boldsymbol{U}_{(t_0)}$ and $\boldsymbol{C}_{(t_0)}$ be the t_0th iteration, $\boldsymbol{U}_{(t_1)}$ and $\boldsymbol{C}_{(t_1)}$ be the t_1th iteration and $f(\boldsymbol{U}, \boldsymbol{C})$ be the objective optimization function.
$\because f(\boldsymbol{U}, \boldsymbol{C})$ is a convex function
$\therefore f(\boldsymbol{U}, \boldsymbol{C})$ must exist a global minimum.
Suppose $t_1 > t_0$ and $f(\boldsymbol{U}_{(t_0)}, \boldsymbol{C}_{(t_0)})$=$f(\boldsymbol{U}_{(t_1)}, \boldsymbol{C}_{(t_1)})$.
From **Algorithm 1**, it is known that there must be $f(\boldsymbol{U}_{(t_0)}, \boldsymbol{C}_{(t_0)}) > f(\boldsymbol{U}_{(t_1)}, \boldsymbol{C}_{(t_1)})$ if **Algorithm 1** executes from t_0 iteration to t_1 iteration.
$\therefore$ The conclusion contradicts the known conditions.
$\therefore$ **Algorithm 1** can converge within finite iterations. □

IV. EXPERIMENTS AND RESULTS

In this section, in order to test the performance of the proposed algorithm, we choose LKPE [26], rLPP [27] and FONPE [28] as comparison algorithm. LKPE, rLPP, FONPE and NGEC are executed on 6 real social network data sets[1]. All algorithms are executed on Python environment. The details of data sets are showed as Table I.

A. Evaluation criteria

In order to effectively evaluate the performance of the algorithms, Fowlkes and Mallows index, Kulczyniski index, Recall, F-measure, Normalized Mutual Information [29] and Modularity [30] are used as evaluation criteria [31]. The evaluation criteria are defined as follows.

(1) Fowlkes and Mallows index:

$$FM = \frac{TP}{\sqrt{(TP + FN)(TP + FP)}} \tag{22}$$

(2) Kulczyniski index:

$$K = \frac{1}{2} \cdot \left(\frac{TP}{TP + FP} + \frac{TP}{TP + FN}\right) \tag{23}$$

(3) Recall index:

$$recall = \frac{TP}{TP + FN} \tag{24}$$

[1] http://www-personal.umich.edu/~mejn/netdata/

TABLE II
THE TEST RESULTS OF THE FOUR ALGORITHMS ON ADJNOUN DATA SET

	LKPE	rLPP	FONPE	NGEC
FM	0.6735	0.6976	0.4960	**0.6976**
K	0.7051	0.7387	0.4960	**0.7387**
recall	0.9141	0.9815	0.4964	**0.9815**
F	0.6432	0.6588	0.4960	**0.6588**
NMI	0.7986	0.9312	0.6698	**0.9312**
Q	0.1531	0.2398	0.1261	**0.2434**
Time	**7.2288**	10.4901	25.4774	124.4427

(4) F-measure:

$$F = \frac{2TP^2}{2TP^2 + TP \cdot FN + TP \cdot FP} \tag{25}$$

where *TP* is the number of pairs of data points which are in the same cluster and also belong to the same class; *FP* is the number of pairs of data points which are in the same cluster but belong to different classes; *FN* is the number of pairs of data points which are in different clusters but belong to the same classes; *TN* is the number of pairs of data points which are in different clusters and also belong to different classes.
(5) Normalized Mutual Information:

$$IN = \frac{\sum_{i=1}^{k}\sum_{j=1}^{|\boldsymbol{Y}|} N_{i,j} \log\left(N \cdot N_{i,j}/N_i \cdot N_j\right)}{\sqrt{\sum_{i=1}^{k} N_i \log\frac{N_i}{N} \cdot \sum_{j=1}^{|\boldsymbol{Y}|} N_j \log\frac{N_j}{N}}} \tag{26}$$

$$NMI = \frac{1}{1+\exp(-IN))} \tag{27}$$

where $N_{i,j}$ is the number of agreements between the *i*th cluster and the *j*th class; N_i is the number of data points in the *i*th cluster; N_j is the number of data points in the *j*th class.
(6) Modularity:

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right) \cdot \delta_{i,j} \tag{28}$$

where A_{ij} is an element of the adjacency matrix of a graph, k_i is the degree of the node v_i and *m* is the number of edges. $\delta_{i,j} = 1$ if the node v_i and v_j are in the same community; otherwise, $\delta_{i,j} = 0$.

For the above evaluation criteria, $0 \leq$ *FM, K, recall, F, NMI* ≤ 1 and $-0.5 \leq Q \leq 1$; a large value means a good result.

B. Experiments and analysis

In order to test the performance of NGEC algorithm, we choose LKPE, rLPP and FONPE as comparison algorithm. All algorithms are tested on the experimental data sets. The number of clusters *k* is set to the real number of communities; if the number of communities is unknown, *k* is set to 2 and $\rho \in \{100, 150, 200, 500\}$. For the parameter *d*, *d* is set to $\text{int}(\min(0.6 \times col, 10 \times k))$ where *col* is the number of nodes. The test results of the algorithms on the experimental data sets are showed as Tables II-VII and Figs.2-4.

TABLE III
THE TEST RESULTS OF THE FOUR ALGORITHMS ON POLITICS DATA SET

	LKPE	rLPP	FONPE	NGEC
FM	0.5129	0.5685	0.5997	**0.6223**
K	0.5131	0.5970	0.6550	**0.6877**
recall	0.5257	0.7793	0.9179	**0.9805**
F	0.5128	0.5413	0.5492	**0.5630**
NMI	0.6412	0.6615	0.7917	**0.9534**
Q	**0.2486**	0.2032	0.2349	0.2454
Time	**5.6923**	15.2529	22.8943	88.1050

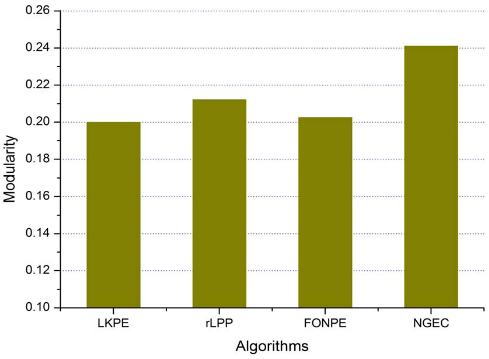

Fig. 2. The Modularity of the four algorithms on Les Miserables data set

TABLE IV
THE TIME CONSUMPTION OF THE FOUR ALGORITHMS ON LES MISERABLES DATA SET

	LKPE	rLPP	FONPE	NGEC
Time	**10.0086**	18.6761	14.9879	59.8132

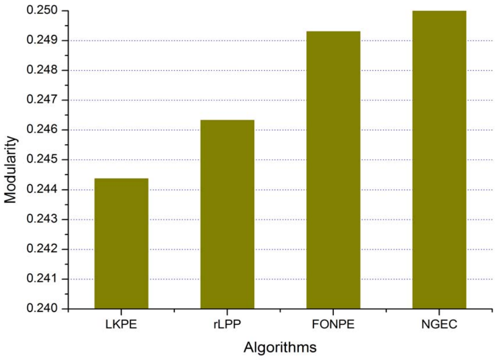

Fig. 3. The Modularity of the four algorithms on cumulative data set

TABLE V
THE TIME CONSUMPTION OF THE FOUR ALGORITHMS ON CUMULATIVE DATA SET

	LKPE	rLPP	FONPE	NGEC
Time	**24.5204**	74.7353	144.7723	3325.6092

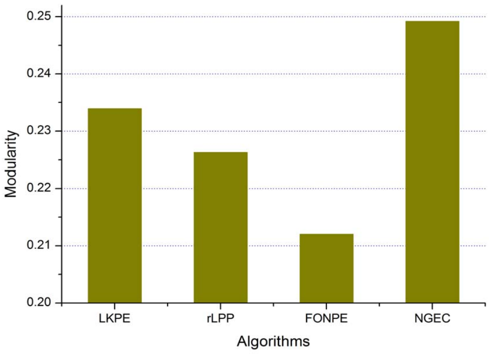

Fig. 4. The Modularity of the four algorithms on Neural network data set

TABLE VI
THE TIME CONSUMPTION OF THE FOUR ALGORITHMS ON NEURAL NETWORK DATA SET

	LKPE	rLPP	FONPE	NGEC
Time	**16.7690**	57.5893	110.4933	2174.4574

Tables II-III are the test results of the four algorithms on the labeled data sets. From the result of Table II, it can be known that NGEC outperforms the comparison algorithms on FM , K, recall, F, NMI and Q. The performance of NGEC is the best. From the result of Table II, NGEC gets the best results on FM , K, recall, F and NMI; the Q index of LKPE is the best of all; however, the Q index of NGEC is the second best and the difference of Q index between LKPE and NGEC is only 0.0032 which is not very obvious. Modularity is an effective evaluation index for the community discovery in unlabeled social network. Figs.2-4 are the results of the four algorithms on the unlabeled social network data sets. From the results, it can be seen that NGEC outperforms the comparison algorithms on Les Miserables, cumulative and Neural network data sets. Tables IV-VII show the time consumption of the four algorithms on Les Miserables, cumulative and Neural network data set and Tables II-III are also showed the time consumption of the four algorithms. From the results, it is known that the time consumption of NGEC is the largest of all. In NGEC algorithm, it uses neighborhood dependence to evaluate the significance of a node for anther node; the complexity of the computing neighbourhood dependence degree is O $\left(n^3\right)$; the complexity of the graph embedding method is O $\left(n^2\right)$ and the complexity of the clustering algorithm is O (n), therefore the complexity of NGEC is O $\left(n^3\right)$ which a high complexity. The results also show that the time performance of NGEC has no advantages comparing with the comparison algorithms.

In order to test the scalability of NGEC algorithm, we choose politics and dolphins as the experimental data sets and the parameter d is changed with different values and $d \in (0, 105]$. NGEC algorithm is executed on the experimental data sets. The test results of the algorithms are showed as Figs.5-6 and Table VII.

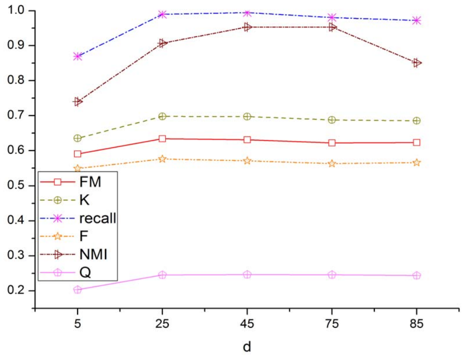

Fig. 5. The test result of NGEC on politics data set

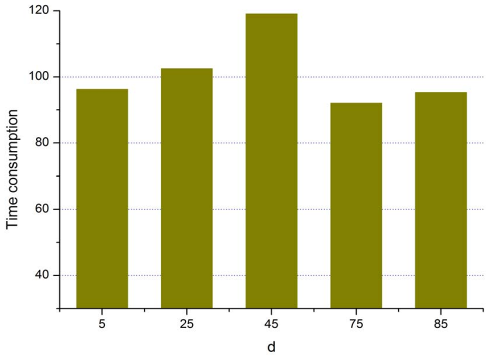

Fig. 6. The time consumption of NGEC on politics data set

Fig.5 and Fig.6 are the test results of NGEC algorithm with different d values. From the results, it is known that d can effect the result of algorithm. On politics data set (Fig.5), the performance of NGEC algorithm improves when d is less than 25; when d is larger than 25, the most evaluation indexes keep stable; NMI is more susceptible to d, relative to the other evaluation indexes. Fig.6 shows the time consumption of NGEC algorithm on politics data set. From the steps of NGEC algorithm, it is known that d has no effect on the calculation of neighborhood dependence and the dimension reduction of graph embedding; d can only effect the nodes clustering. However, most time is spent on the calculation of neighborhood dependence and parallel computation mechanis-

TABLE VII
THE TEST RESULTS OF NGEC ON DOLPHINS DATA SET

d	3	10	20	30	40	50
Q	0.1512	0.2102	0.2021	0.2246	0.2218	0.2183
Time	52.5630	44.3925	43.6435	44.2595	46.4757	43.6735

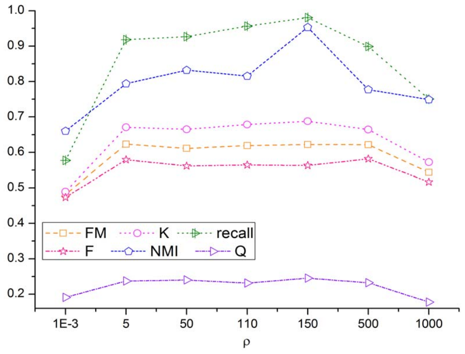

Fig. 7. The test result of NGEC on politics data set with different ρ values

m is adopted in the clustering algorithm which decreases the significance of the effect; therefore the effect of d on the time consumption of NGEC algorithm is not significant. Table VII shows the test results of NGEC algorithm with different d values on dolphins data set. From the results, it can be seen that the Q value improves when d is less than 30, then the change of the Q value is not significant. The fluctuation of time consumption of NGEC algorithm on dolphins data set is also small which is consistent with the conclusion of NGEC algorithm on politics data set.

In order to test the effect of ρ on the performance of NGEC algorithm, we choose politics as experimental data set. NGEC algorithm is executed on the data set with different ρ values. The test result is showed in Fig.7.

Fig.7 shows the test result of NGEC algorithm on politics data set with different ρ values. From Fig.7, it is known that ρ has an effect on the performance of NGEC algorithm. When ρ is not too large, the performance of NGEC algorithm improves with the increase of ρ value. When $\rho = 150$, the performance of NGEC algorithm is the best on politics data set. However, when ρ is larger than 150, the performance of NGEC algorithm decreases. From Eq.(21), it is known that ρ can balance the weight entropy and the error of clustering result; a larger ρ will decrease the influence of distance over the degree of membership; if $\rho \to \infty$, the distance has no function for the degree of membership and all data points will be partitioned into the same cluster which is a bad result. Therefore the above result shows that a too large ρ has a negative impact on the clustering result of NGEC algorithm and a reasonable ρ is important for NGEC algorithm.

V. Conclusions

In this paper, a neighborhood graph embedding for nodes clustering (NGEC) of social network is proposed. NGEC employs neighborhood dependence to determine the dependence of a node for target node. The matrix of dependence degree is as an new presentation of a graph. Considering of the high dimension of the matrix of dependence degree, we introduce a neighborhood graph embedding method to decrease the dimension and then propose a new clustering algorithm which can obtain a minimum error and minimum weight entropy to complete nodes clustering. NGEC algorithm and the comparison algorithms are executed on the real social network data sets and uses 6 evaluation indexes to evaluate the results of the algorithms. The experimental results show that NGEC algorithm outperforms the comparison algorithms which means that NGEC is an effective algorithm for community discovery.

However, there are still some problems which should be further researched. For NGEC algorithm, it can only discover non-overlapping communities. In fact, overlapping communities are common in real applications. In addition, it is known from the experiments that the time complexity of NGEC algorithm is higher than the comparison algorithms; therefore how to extend NGEC algorithm to overlapping communities and decrease the time complexity of NGEC algorithm will be our research directions in the future.

Acknowledgment

This work was supported by National Natural Science Fund of China (Nos.61672130, 61602082, 61627808, 91648205), MOE Research Center for Online Education of China (No.2016YB121), the Fundamental Research Funds for the Central Universities (No.DUT17RC(3)071), the Open Program of State Key Laboratory of Software Architecture (No.SKLSAOP1701), Liaoning Revitalization Talents Program (XLYC1806006), and the development of science and technology of Guangdong province special fund project (No.2016B090910001).

References

[1] R. Wang, X. Jia, Q. Li, S. Zhang, Machine learning based cross-site scripting detection in online social network, in: 2014 IEEE Intl Conf on High Performance Computing and Communications (HPCC), IEEE, 2014, pp. 823–826.

[2] J. Kim, M. Hastak, Social network analysis: Characteristics of online social networks after a disaster, International Journal of Information Management 38 (1) (2018) 86–96.

[3] C. Zang, P. Cui, C. Faloutsos, Beyond sigmoids: The nettide model for social network growth, and its applications, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 2015–2024.

[4] K. Ye, X. Jiang, S. Chen, D. Huang, B. Wang, Analyzing and modeling the performance in xen-based virtual cluster environment, in: 2010 IEEE 12th International Conference on High Performance Computing and Communications (HPCC), IEEE, 2010, pp. 273–280.

[5] S. P. Borgatti, A. Mehra, D. J. Brass, G. Labianca, Network analysis in the social sciences, science 323 (5916) (2009) 892–895.

[6] C. C. Aggarwal, An introduction to social network data analytics, in: Social network data analytics, Springer, 2011, pp. 1–15.

[7] L. Liao, X. He, H. Zhang, T.-S. Chua, Attributed social network embedding, IEEE Transactions on Knowledge and Data Engineering 30 (12) (2018) 2257–2270.

[8] B. Bilecen, M. Gamper, M. J. Lubbers, The missing link: Social network analysis in migration and transnationalism, Social Networks 53 (2018) 1–3.

[9] L. Hutton, T. Henderson, Toward reproducibility in online social network research, IEEE Transactions on Emerging Topics in Computing 6 (1) (2018) 156–167.

[10] B. Boden, S. Günnemann, T. Seidl, Tracing clusters in evolving graphs with node attributes, in: Proceedings of the 21st ACM international conference on Information and knowledge management, ACM, 2012, pp. 2331–2334.

[11] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, Density-based spatial clustering of applications with noise, in: Int. Conf. Knowledge Discovery and Data Mining, Vol. 240, 1996.

[12] Y. Qian, Y. Li, M. Zhang, G. Ma, F. Lu, Quantifying edge significance on maintaining global connectivity, Scientific reports 7 (2017) 45380.

[13] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[14] Z. Lu, X. Sun, Y. Wen, G. Cao, T. La Porta, Algorithms and applications for community detection in weighted networks, IEEE Transactions on Parallel and Distributed Systems 26 (11) (2015) 2916–2926.

[15] C. Liu, J. Liu, Z. Jiang, A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks, IEEE transactions on cybernetics 44 (12) (2014) 2274–2287.

[16] J. J. Whang, D. F. Gleich, I. S. Dhillon, Overlapping community detection using neighborhood-inflated seed expansion, IEEE Transactions on Knowledge and Data Engineering 28 (5) (2016) 1272–1284.

[17] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web., Tech. rep., Stanford InfoLab (1999).

[18] W. Liu, X. Jiang, M. Pellegrini, X. Wang, Discovering communities in complex networks by edge label propagation, Scientific reports 6 (2016) 22470.

[19] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical review E 76 (3) (2007) 036106.

[20] J. Eustace, X. Wang, Y. Cui, Community detection using local neighborhood in complex networks, Physica A: Statistical Mechanics and its Applications 436 (2015) 665–677.

[21] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in neural information processing systems, 2002, pp. 849–856.

[22] Y. Li, K. He, D. Bindel, J. E. Hopcroft, Uncovering the small community structure in large networks: A local spectral approach, in: Proceedings of the 24th international conference on world wide web, International World Wide Web Conferences Steering Committee, 2015, pp. 658–668.

[23] D. F. Gleich, C. Seshadhri, Vertex neighborhoods, low conductance cuts, and good seeds for local community methods, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2012, pp. 597–605.

[24] W. Liu, M. Pellegrini, X. Wang, Detecting communities based on network topology, Scientific reports 4 (2014) 5739.

[25] X. He, P. Niyogi, Locality preserving projections, in: Advances in neural information processing systems, 2004, pp. 153–160.

[26] A. Elbagoury, R. Ibrahim, M. S. Kamel, F. Karray, Ebek: Exemplar-based kernel preserving embedding., in: International Joint Conferences on Artificial Intelligence, 2016, pp. 1441–1447.

[27] H. Wang, F. Nie, H. Huang, Learning robust locality preserving projection via p-order minimization, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 3059–3065.

[28] T. Pang, F. Nie, J. Han, Flexible orthogonal neighborhood preserving embedding., in: International Joint Conferences on Artificial Intelligence, 2017, pp. 2592–2598.

[29] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recognition 43 (3) (2010) 767–781.

[30] M. E. Newman, Communities, modules and large-scale structure in networks, Nature physics 8 (1) (2012) 25–31.

[31] J. Xie, Unsupervised Learning Methods and Applications, Publishing Hourse of Electronics Industry, 2016.