Non-negative Matrix Factorization with Community Kernel for Dynamic Community Detection

Saisai Liu and Zhengyou Xia(✉)

College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
zhengyou_xia@nuaa.edu.cn

Abstract. Finding community structures from user data has become a hot topic in network analysis. However, there are rarely effective algorithms about the dynamic community detection. To recalculate the whole previous nodes and deal with excessive calculation is used to solve the problem of dynamically adding community nodes in the previous researches. In this paper, we propose an incremental community detection algorithm without recalculating the whole previous nodes using the incremental non-negative matrix factorization (INMF). In this algorithm, community kernel nodes with the largest node degree and adjacent triangle ratio is selected to calculate the data feature matrix, then the complexity of the calculations is largely simplified by reducing the dimension of the data feature matrix. We also propose a strategy to solve the problem of ensuring the feature space dimension and community number of NMF. We discuss our method with several previous ones on real data, and the results show that our method is effective and accurate in find potential communities.

Keywords: Community detection · Community kernel · Data mining · INMF · NMF

1 Introduction

Many systems take the form of networks. A complex network is composed of a large number of nodes and intricate relationships between them [1, 2]. Most networks seem to share a number of distinctive statistical properties, such as the small world effect, network transitivity, community structure, and so on. A community is a subset of these nodes. The connections are frequent in the same community and sparse between different communities [1, 3–5].

Non-negative matrix factorization (NMF) reduces the matrix dimension by factorizing a Non-negative data feature matrix into two matrices [6]. All the factor matrices are restricted to be Non-negative. The factorization can also make the characteristics of the data more obvious, which is conducive to extract data features. There are some application problems of NMF. One is the high computational complexity [6]. Another problem is the difficulty of determining the feature space dimension.

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 778–792, 2019.
https://doi.org/10.1007/978-3-030-32388-2_64

In this paper, we proposed a incremental Non-negative matrix factorization algorithm with community kernel (CK-INMF) for dynamic community detection. The method can effectively solve above problems by using the community kernels to reduce the dimension of the data feature matrix and avoid the discussion of dimensions by clustering.

Most of the existing community detection algorithms are about static community. During the execution process of those algorithms, new nodes cannot be dynamically added. Dynamic community detection requires the ability to dynamically add some nodes. In most algorithms, adding new nodes requires to abandon the existing result of previous nodes and recalculate from the very beginning of the algorithm. It increases the computational cost.

CK-INMF (our method) expanses the current nodes feature matrix dimension and calculate the result base on the previous factorization. It offered an effective dynamic community strategy.

The remainder of the paper is organized as follows. Section 2 introduces the related work. We explain our CK-INMF model in Sect. 3. Then simulation experiments are carried out in Sect. 4. Finally, a concise conclusion is given in Sect. 5.

2 Related Work

Community detection has become a research hot spot in recent years, and more scholars have devoted to the research of community detection algorithm. In 2003, Newman first proposed the concept of modularity Q in community detection and achieved the effect of community division by optimizing the value of Q [7]. At present, common community detection algorithms can be divided into some categories, including graph-based segmentation, objective function optimization, clustering heuristic method, and so on [8–11]. The usual community detection studying belongs to the field of unsupervised learning. In recent years, scholars have paid attention to apply the Non-negative matrix factorization (NMF) to the community detection for its successful application in unsupervised learning.

NMF is a matrix decomposition strategy, first proposed by Paatero and Tapper [12]. NMF is an efficient clustering tool for small data sets. The NMFIB (Non-negative matrix factorization with iterative bipartition) proposed by He can obtain more statistically significant GO (geneontology) in the protein interaction network [12]. The BNMF (Bayesian Non-negative matrix factorization) model proposed by Rsorakis can obtain a more accurate community structure comparing with traditional hierarchical and spectral clustering methods [13].

Our algorithm CK-INMF is an improvement of algorithms based on NMF. Current NMF community detection algorithms usually adopt method of factorize a Non-negative data feature matrix $V = (v_1, v_2, \ldots, v_n)$ into the product of the feature space basis matrix W, the encoding matrix H, and $V_{mn} \approx W_{ms}H_{sn}$ shown in Fig. 1.

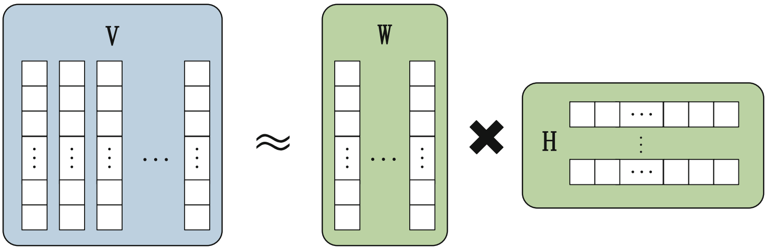


Fig. 1. NMF factorization formula V ≈ WH

Each column of the data feature matrix V represents a different node, and each column in V corresponds to one specific column of the encoding matrix H. s represents the new feature space dimension and the number of communities.

The goal of the NMF is to minimize the reconstruction error between the matrix V and WH. The objective function F is given by (1).

$$F(W,H) = \frac{1}{2}\|V - WH\|^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(V_{ij} - (WH)_{ij})^2 \tag{1}$$

To optimize the problem of (1), iterative multiplicative (2), (3) can be used to update W and H during calculation.

$$W_{i\mu} \leftarrow W_{i\mu}\frac{(VH^T)_{i\mu}}{(WHH^T)_{i\mu}} \tag{2}$$

$$H_{\mu j} \leftarrow H_{\mu j}\frac{(W^TV)_{\mu j}}{(W^TWH)_{\mu j}} \tag{3}$$

$$\text{s.t.}\ i = 1,\ldots,m; j = 1,\ldots,n; \mu = 1,\ldots,s$$

According to the encoding matrix H, we can get the detected communities. h_{ij} reflects the extent of i_{th} node belonging to the j_{th} community. Let $h_{ik} = \max\{h_{i1}, h_{i2}, \ldots, h_{is}\}$. If the i_{th} node is divided into the only k_{th} community, the non-overlapping communities can be got; if the i_{th} node is divided into several communities, overlapping communities can be got.

There are some problems of current NMF community detection algorithms, as following:

(1) Most algorithms adopts the adjacency matrix as the data feature matrix V, and the high dimensional V increase the computational complexity.

(2) The new feature space dimension s has a great impact on the performance of community detection. But it is difficult to decide the precise value of s.

(3) When apply NMF to the dynamic community, it should recalculate the whole matrices. The cost is huge.

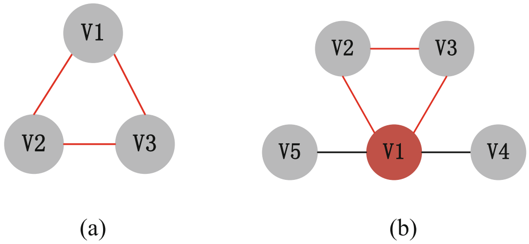
To deal with these problems, we propose the CK-NMF algorithm.

3 Our Algorithm CK-INMF: Community Detection Algorithm Model Based on INMF and Community Kernel

To dynamically add nodes, reduce the computational complexity and solve the problem of determining the feature space dimension during factorization, we suppose the CK-INMF algorithm. In CK-INMF, we use the INMF expanded NMF to detect the dynamical community. In this section, we would give the detail description of our CK-INMF algorithm. INMF is often used to the online blind source separation [9] and image and video processing [14]. Here we apply INMF to achieve the dynamic community detection. CK-INMF would adopt community kernel nodes to reduce the dimension of data feature matrix V. We cluster the preliminary detected communities to get the final detection, solving the problem of deciding the value of parameter s.

3.1 Detecting the Community Kernel Nodes and Reducing the Dimension of Feature Matrix

In CK-NMF algorithm, the initial Matrix Vmn is assigned based on the community kernel. Therefore, the first step of our algorithm is to find the community kernel. The community kernel is a set of nodes with greater influence than the other nodes in a network [15]. We believe that nodes with big degree and structure shown in Fig. 2 are more likely to become community kernel nodes. Figure 2(a) is an triangle structure. Node v1 own more influence than others and is one of the community kernel nodes in Fig. 2(b). So we measure the influence of one node by the degree of it and the ratio of the triangles that can be formed with its adjacent nodes.



(a) (b)

Fig. 2. Triangle structure in networks

The community kernel set includes m most influential nodes. CK-INMF adopts the matrix V_{mn} instead of V_{nn} and m ≪ n. So the computation complexity of the whole algorithm is greatly reduced. We select m nodes with the highest kernel weights. The weight formulation is (4):

$$w_i = r\left(\frac{den(i)}{\max Den}\right) + (1-r)\left(\frac{2triangle(i)}{den(i)(den(i)-1)}\right) \tag{4}$$

$$\text{s.t.}\, den(i) = \sum\nolimits_j A_{ij}\ maxDen = max\{den(1), den(2), \ldots, den(n)\}$$

$$triangle(i) = \sum\nolimits_{j,h} a_{ij}\, a_{ih}\, a_{jh}$$

The detail description of community kernel detection is shown below.

```
**Algorithm1  Community kernel detection algorithm**
**Input**: Graph G=(V,E), size of kernel m
**Output**: A set of community kernel $K=(k_2,k_1,\ldots,k_m)$
$K \leftarrow \phi$
calculate the kernel weight of all nodes by (4)
all nodes are ordered in a descending influence sequence and stored in queue D
store the selected nodes during every cycle in queue stepK
/* Find m community kernel nodes */
**While** |K|<m **do**
    $stepK \leftarrow \phi$
    **While** (|K|+|stepK|)<m **do**
        /*store the selected nodes during this cycle in stepK*/
        obtain the front v from queue (D-K-stepK);
        **if** $\neg\exists edge(v,k_i),\forall k_i \in stepK$ then
            assign v as the kernel node: $stepK \leftarrow \{v\}$
    K=K+stepK
return community kernels $K=(k_1,k_2,\ldots,k_m)$
```

In CK-INMF, we adopt adjacency matrix A_{nn} of the network as V, but we would use the community kernel to reduce the dimension of $A = \{a_1^T, a_2^T, \ldots, a_n^T\}$. We delete the redundant rows $a_r = \{r \notin K, K = \{k_1, k_2, \ldots, k_m\}$ from A, and get matrix V, as shown in Fig. 3.

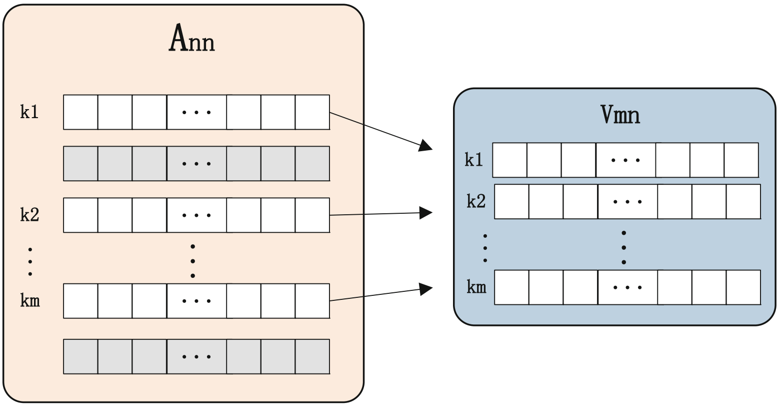


Fig. 3. Calculate matrix V according to adjacent matrix A

3.2 Dynamically Adding New Nodes

In some cases, we need to add some new nodes to the existing network. The goal of INMF equations are to update the matrices W and H while new nodes are added to the existing network. There is no need to recalculate the previous nodes by INMF. During this process, V and H will be added some new columns and W would be updated.

Let W_k and H_k represent the decomposition matrices obtained from the initial k nodes (k $\geq$ 2 m), then the objective function of Non-negative matrix factorization can be expressed as (5).

$$F_{\mathrm{k}} = \|V - WH\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \left(V_{ij} - (W_k H_k)_{ij}\right)^2 \tag{5}$$

When the V_{k+1} nodes is added, both W and H are changed and the reconstructed objective function can be expressed as (6).

$$\begin{aligned} F_{k+1} &= \|V' - W_{k+1}H_{k+1}\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{k+1} \left(V_{ij} - (W_{k+1}H_{k+1})_{ij}\right)^2 \\ &= \sum_{i=1}^{m} \sum_{j=1}^{k} \left(V_{ij} - (W_{k+1}H_{k+1})_{ij}\right)^2 + \sum_{i=1}^{m} \left((v_{k+1})_i - (W_{k+1}H_{k+1})_i\right)^2 \\ &\cong F_k + f_{k+1} \end{aligned} \tag{6}$$

From above, we can find that when the sample increases, the new objective function is the sum of the objective function F_k and increment f_{k+1}. In this way, the objective function F can be updated based on newly added sample without recalculating.

With the INMF in incremental learning, the variables of the objective function are the column and the base matrix. Using the gradient descent method, the iteration rules are as (7), (8).

$$(h_{k+1})_{\mu} \leftarrow (h_{k+1})_{\mu} \frac{(W_{k+1}^T v_{k+1})_{\mu}}{(W_{k+1}^T W_{k+1} h_{k+1})_{\mu}} \tag{7}$$

$$(W_{k+1})_{i\mu} \leftarrow (W_{k+1})_{i\mu} \frac{(V_{k+1} H_{k+1}^T + v_{k+1} h_{k+1}^T)_{i\mu}}{(W_{k+1} H_k H_k^T + W_{k+1} h_{k+1} h_{k+1}^T)_{i\mu}} \tag{8}$$

$$s.t.\ \mu = 1, \ldots, n; i = 1, \ldots, s$$

Besides saving the current iterative values h and W after every iteration, history information also needs to be stored for the next update. The storage matrices are shown in (9), (10).

$$A = V_{k+1} H_{k+1}^T = V_k H_k^T + v_{k+1} h_{k+1}^T \tag{9}$$

$$B = H_{k+1} H_{k+1}^T = H_k H_k^T + h_{k+1} h_{k+1}^T \tag{10}$$

From the iterative equations, we can infer that the storage matrices could reduce the amount of calculation during every iteration.

The process of adding new nodes is shown in Fig. 4.

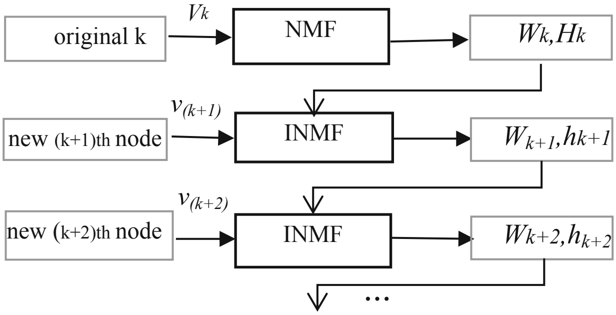


Fig. 4. The process of CK-INMF adding new nodes

3.3 Community Detecting Based on Encoding Matrix H

The factorization result is $V_{mn} \approx W_{ms}H_{sn}$. In current NMF non-overlapping community detection, let $h_{ik} = \max\{h_{i1}, h_{i2}, \ldots, h_{is}\}$ and the i_{th} node is mapped to the only k_{th} community. The s represent the community number, but sometimes we do not know the community number. So we need to search the value by a lot of test. But it is hard to decide the precise value of the community number. In CK-INMF, we propose a strategy to solve it by select a integer as s, which is larger than the ideal community number, then cluster the s communities into smaller amount communities according to modularity Q. During the clustering process, we calculate the ΔQ of each pair of communities. The we cluster the two communities which produce the biggest ΔQ, and calculate the ΔQ of new community with others and continue to compare ΔQ. Because of $s \ll n$, the strategy is effective. Figure 5 gives the description of the process of community detecting.

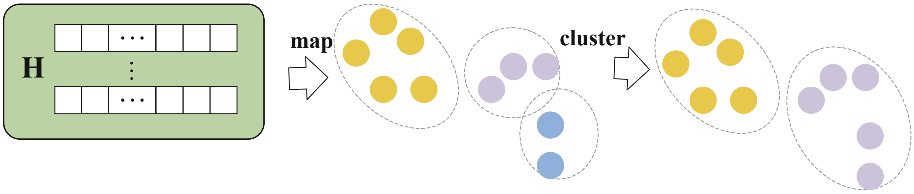


Fig. 5. Detecting communities based on encoding matrix H

3.4 CK-INMF Algorithm Pseudo-code Description

Figure 6 shows the CK-INMF algorithm framework.

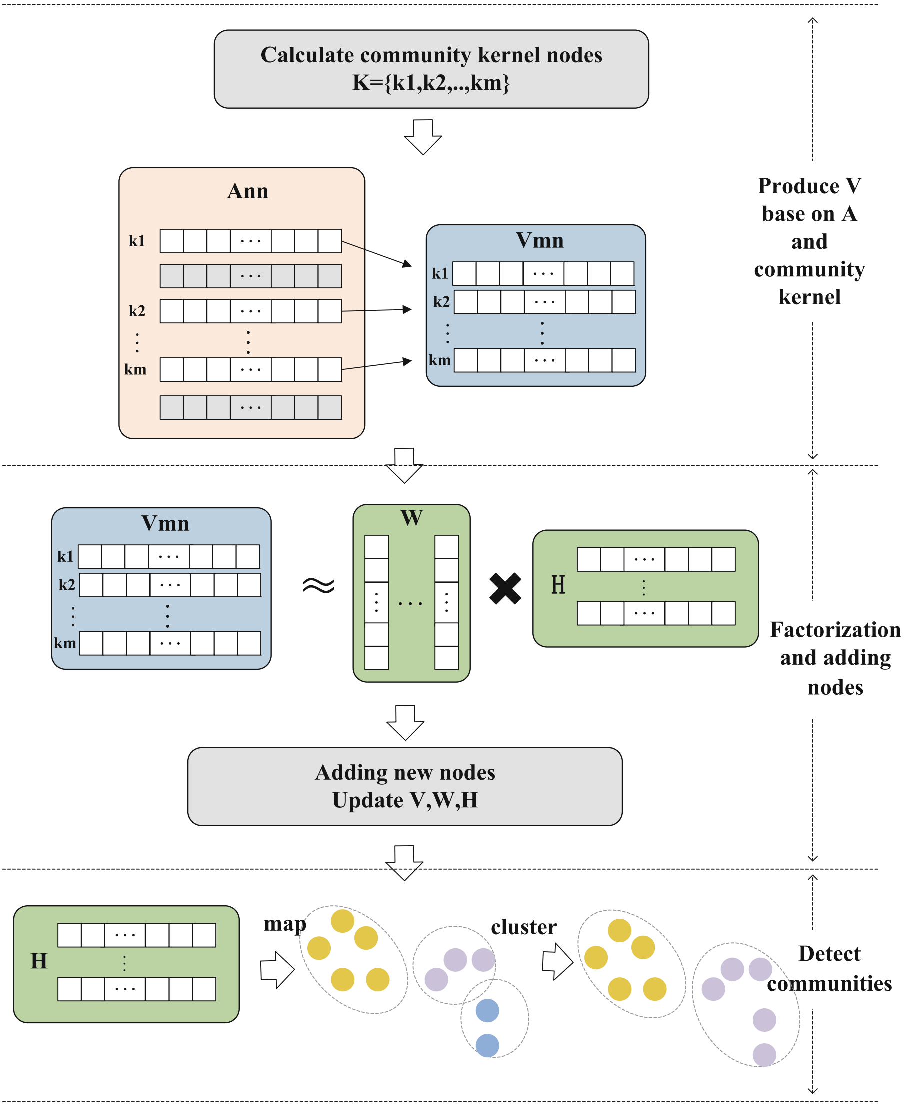


Fig. 6. The CK-INMF framework

The pseudo-code description of CK-INMF is shown below.

```
Algorithm2 CK-INMF
Input: Graph G=(V,E), new nodes set $V_{new}$, feature space dimension s, number of kernel nodes m
Output: A set of community $C=(c_1,c_2,\dots,c_r)$
$K \leftarrow \phi \quad C \leftarrow \phi \quad F=1$
calculate the community kernel $K=(k_1,k_2,\dots,k_m)$ based on algorithm1
calculate the feature matrix $V_{mn}$
for all $v_i \in V$ (i>1) do
    /* update equations iteratively for i=1,..,m;j=1,...,n,u=1,...,s */
    While F>$10^{-3}$ and times<200 do
        $W_{i\mu} \leftarrow W_{i\mu} \frac{(VH^T)_{i\mu}}{(WHH^T)_{i\mu}}$
        $H_{\mu j} \leftarrow H_{\mu j} \frac{(W^T V)_{\mu j}}{(W^T WH)_{\mu j}}$
/* adding new nodes */
for all $v_k \in V_{new}$ (k>1) do
    add the new node $v_k$ as the $k_{th}$ column of V
    /* update equations for i=1,...,m;u=1,...,s*/
    While $f_k$>$10^{-4}$ and times<200 do
        $(h_k)_\mu \leftarrow (h_k)_\mu \frac{(W_k^T v_k)_\mu}{(W_k^T W_k h_k)_\mu}$
        $(W_{k+1})_{i\mu} \leftarrow (W_{k+1})_{i\mu} \frac{(V_k H_k^T + v_{k+1} h_{k+1}^T)_{i\mu}}{(W_{k+1} H_k H_k^T + W_{k+1} h_{k+1} h_{k+1}^T)_{i\mu}}$
    calculate the matrices A,B
/*According to H, divide nodes into m preliminary communities P*/
for all $v_k \in V$ do
```

```
    add $v_k$ to $P_{index}$, $P_{index}(index = rowNumberOf\ \max\{H_{1k}, H_{2k},\dots,H_{sk}\})$
/* cluster the m communities P according to modularity Q*/
for all i,j $\in n$ do
    if q=Q($P_i$)+Q(Pj)-Q($P_i$+$P_j$)>0 do
        add(i,j,q) into Qqueue
While Qqueue!=null do
    sort Qqueue from most to least based on q
    delete the first Qqueue to(i,j,q)
    delete all element of Qqueue including i and j
    merge $P_i$ and $P_j$ to $P_k$ and calculate q with existing communities
    add new q into Qqueue
return the existing communities $C=(c_1,c_2,\dots,c_r)$
```

4 Experimental Result and Analysis

In order to verify the feasibility of our proposed algorithm, we analyze the parameters of r and m by real-world networks dataset. We also compare CK-INMF with other related algorithms by real-world networks dataset. CK-INMF is implemented in python (Python 2.7) language, and the program runs as window 10 operating system, 2.00 GHZ, 8 GB memory.

4.1 Experiment Data and Evaluation Standard

In order to analysis the performances of CK-INMF, three real networks are selected for experiment. The real networks include the classic Zachary's Karate Club Network [16], the dolphin social network [17] and the college football network [18]. The Zachary's Karate Club Network owns a number of 34 nodes and 78 edges. The dolphin social network is a community of 62 nodes and 159 edges. The college football network consists of 115 nodes and 616 edges.

Here we adopt modularity (Q) and normalized mutual information(NMI) as evaluation standards to measure the effectiveness of CK-INMF partitioning result. The larger the values are, the better the results is.

Suppose graph G = (V, E) with N nodes, edges set E, community detection C = [c_1, c_2, .., c_r]:

The modularity Q is defined as follows:

$$Q = \frac{1}{2|E|}\sum\nolimits_{ij}\left(\left(A_{ij} - \frac{k_i\,k_j}{2|E|}\right) \times \delta(\mathrm{c}(i), c(j))\right)$$

$$\text{s.t.}\, k_{\mathrm{i}} = \sum\nolimits_{\mathrm{j}} A_{ij}\,\delta(c(i), c(j)) = \begin{cases} 0, & \mathrm{c}(i)! = c(j) \\ 1, & c(i) = c(j) \end{cases}$$

Given a standard community detection result S = [s_1, s_2, …, s_r]

The NMI is defined as follows:

$$MI(X, Y) = \sum_{i=1}^{|S|}\sum_{j=1}^{|C|} P(i,j)\log\left(\frac{P(i,j)}{P(i)P'(j)}\right)$$

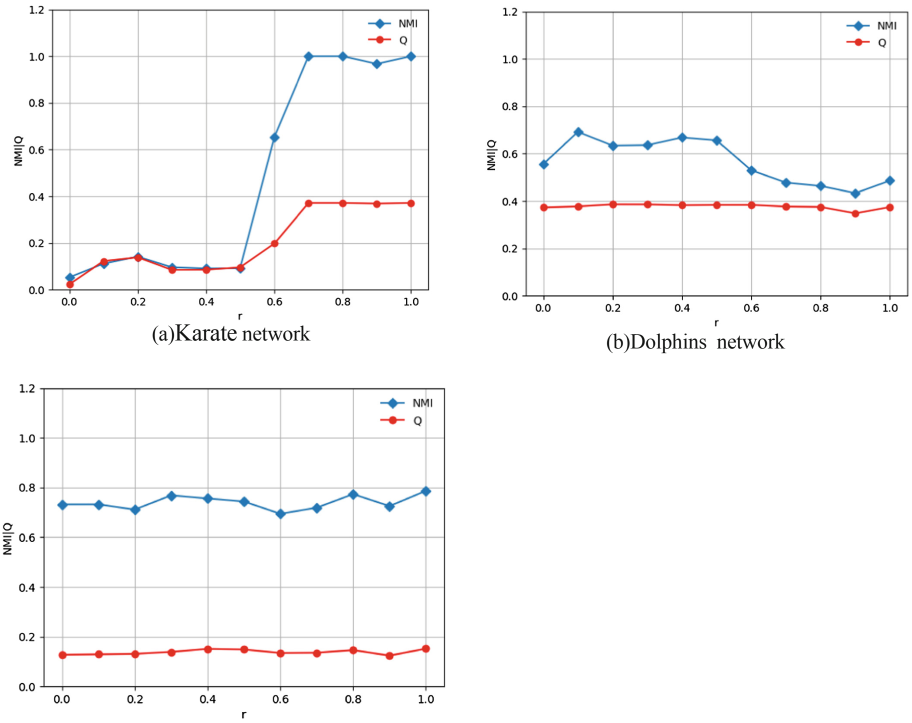
$$\text{s.t.}\, P(i,j) = \frac{|\, S_i \cap C_j)}{N}; P(i) = \frac{|\, S_i\,|}{N}; P'(j) = \frac{|\, C_j\,|}{N}$$

$$NMI(S, C) = \frac{2MI(S, C)}{H(S) + H(C)}$$

$$s.t.\, H(S) = -\sum_{i=1}^{|S|} P(i)\log(P(i)); H(C) = -\sum_{j=1}^{|C|} P'(j)\log(P'(j));$$

4.2 Parameter Analysis of CK-INMF

The performances of CK-INMF are based on the selection of r. The results are shown in Fig. 7. The performance of the experiences change by the value of r. We can adjust the value of based on Q. We usually take the value of r which can maximize the value of Q.



(a)Karate network

(b)Dolphins network

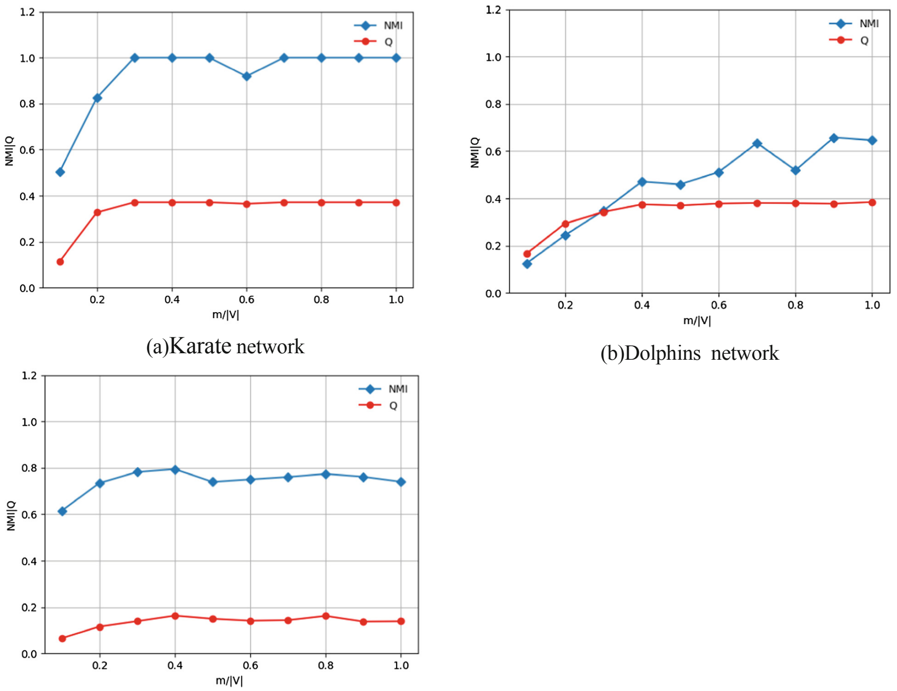
(c)Football network

Fig. 7. The NMI and Q results of CK-INMF on networks with r

The performances of CK-INMF are based on the selection of m. The obtained results are shown in Fig. 8. It tells us that we can select m between 0.3–0.5 of the ratio of m to e, to balance the performance and cost.

4.3 Comparison of CK-INMF with Other Related Algorithms

We compared CK-INMF with NMF_{KL} [3], BNMF [19], $\text{BNMTF}_{\text{LSE}}$ [20], $\text{sBNMTF}_{\text{LSE}}$ [20], SNMF [21]. The results are shown in Table 1. From it we can conclude that our algorithm is great predominant than others on Karate network, and need to improve on Dolphins and Football networks.



(a)Karate network

(b)Dolphins network

(c)Football network

Fig. 8. The NMI and Q results of CK-INMF on networks with m

Table 1. NMI of the community detection algorithms on real networks

Algorithm	Karate	Dolphins	Football
CK-INMF	0.996	0.634	0.794
$\mathrm{NMF}_{\mathrm{KL}}$	0.437	0.775	0.891
BNMF	0.603	0.831	0.878
$\mathrm{BNMTF}_{\mathrm{LSE}}$	0.553	0.590	0.891
$\mathrm{sBNMTF}_{\mathrm{LSE}}$	0.545	0.045	0.462
SNMF	0.983	0.872	0.902

5 Conclusion

We propose an CK-INMF incremental community detection algorithm. It gives a dynamic community detection model of achieving adding nodes without recalculating the whole existing nodes, which could save a lot of time coat. The algorithm firstly selects the m community kernel nodes with the largest node degree and adjacent triangle ratio, then calculates the feature matrix according to the m nodes. Comparing with the other NMF algorithms which directly use the complete adjacency matrix, the calculation complexity is largely simplified by reducing the dimension of the matrix.

By clustering the factorization encoding matrix, the problem of judging the dimension of feature space and the community number is solved. The performance of our algorithm on several real networks shows it is effective and useful. However, the parameter combination selected cannot play the best effect, and it is remained to future work to come up with a better way to balance the combination performance of parameters.

Acknowledgment. This paper is supported by National Natural Science Foundation of China (71871109).

References

1. Bu, Z., Zhang, C., Xia, Z., Wang, J.: A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network. Knowl.-Based Syst. **50**(Complete), 246–259 (2013)
2. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **78**(4 Pt 2), 046110 (2008)
3. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
4. Li, H.J., Bu, Z., Li, A., Liu, Z., Yong, S.: Fast and accurate mining the community structure: integrating center locating and membership optimization. IEEE Trans. Knowl. Data Eng. **28**(9), 2349–2362 (2016)
5. Zhan, B., Wu, Z., Jie, C., Jiang, Y.: Local community mining on distributed and dynamic networks from a multiagent perspective. IEEE Trans. Cybern. **46**(4), 986–999 (2017)
6. Zhan, B., Xia, Z., Wang, J., Zhang, C.: A last updating evolution model for online social networks. Phys. A Stat. Mech. Appl. **392**(9), 2240–2247 (2013)
7. Newman, M.E.: Fast algorithm for detecting community structure in networks. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **69**(6 Pt 2), 066133 (2004)
8. Bucak, S.S., Gunsel, B.: Incremental subspace learning via non-negative matrix factorization. Pattern Recognit. **42**(5), 788–797 (2009)
9. Guoxu, Z., Zuyuan, Y., Shengli, X., Jun-Mei, Y.: Online blind source separation using incremental nonnegative matrix factorization with volume constraint. IEEE Trans. Neural Netw. **22**(4), 550–560 (2011)
10. Adamic, L.A.: The political blogosphere and the 2004 u.s. election: Divided they blog. In: International Workshop on Link Discovery (2005)
11. Andrea, L., Santo, F., Filippo, R.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E: Stat., Nonlin. Soft Matter Phys. **78**(4 Pt 2), 046110 (2008)
12. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2), 111–126 (2010)
13. Dongxiao, H., Di, J., Carlos, B., Dayou, L.: Link community detection using generative model and nonnegative matrix factorization. PLoS ONE **9**(1), e86899 (2014)
14. Bucak, S.S., Gunsel, B.: Incremental clustering via nonnegative matrix factorization. In: International Conference on Pattern Recognition (2012)
15. Zhen, L., Zheng, X., Nan, X., Chen, D.: CK-LPA: efficient community detection algorithm based on label propagation with community kernel. Phys. A Stat. Mech. Appl. **416**(C), 386–399 (2014)
16. Wayne, Zachary: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**(4), 452–473 (1977)

17. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav. Ecol. Sociobiol. **54**(4), 396–405 (2003)
18. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. U.S.A. **99**, 7256–7821 (2002)
19. Ioannis, P., Stephen, R., Mark, E., Ben, S.: Overlapping community detection using bayesian non-negative matrix factorization. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **83**(2), 066114 (2011)
20. Yu, Z., Yeung, D.Y.: Overlapping community detection via bounded nonnegative matrix tri-factorization. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2012)
21. Fei, W., Ding, C.: Community discovery using nonnegative matrix factorization. Data Min. Knowl. Discovery **22**(3), 493–521 (2011)