2017 2nd International Conference on Artificial Intelligence and Engineering Applications (AIEA 2017)
ISBN: 978-1-60595-485-1

Community Detection of Heterogeneous Networks with RankClus Based on Spark GraphX

YU YANG, YONG GUO, BO DENG and HAILONG LI

ABSTRACT

Most real systems consist of a large number of interacting, multi-typed components, while most contemporary researches model them as homogeneous networks without distinguishing different types of objects and links in the networks. Compared with the homogeneous networks, community detection based on the heterogeneous networks could obtain more accurate community structure. Most of contemporary community detection tasks only work on small dataset and fail to consider the quick and parallel process on big data. In this paper, we propose a method to detect communities in heterogeneous networks and solve the parallelisim problem on big networked data. Firstly, we improve and optimize the RankClus algorithm for community detection in heterogeneous networks. Then, we parallelize the improved algorithm based on spark graphx. Finally, we present the experiments of the parallel algorithm and compare the different experiment results which demenstrates that our method is applicable and has efficient results for community detection of heterogeneous networks.

KEYWORDS

Heterogeneous networks, Community detection, Parallelization.

INTRODUCTION

We know that most real systems usually consist of a large number of interacting, multi-typed components, such as human social activities, communication and computer systems, and biological networks. In the real world, a lot of complicated systems could be modelled as information networks with the nodes and edges representing the entities and the relationships between the entities of the systems. So we will have a better understanding of the real systems by researching the information networks. Information networks are ubiquitous and form a critical component of modern information infrastructure. For example, the size of social networks have been increasing more and more quickly in recent years. According to the latest report [1], the global internet users increased 332 million and reached 3.42 billion, and it increased the number by 10% and equal to 46% of the world's population.

Yu Yang, Beijing Institute of System Engineering, Beijing 100101, China; fqyang_y@163.com.
Yong Guo, Beijing Institute of System Engineering, Beijing 100101, China; fqyang11@sina.com.
Bo Deng Beijing Institute of System Engineering, Beijing 100101, China; deng@163.com.
Hailong Li, Beijing Institute of System Engineering, Beijing 100101, China; lihlong@163.com.

There is a large amount of information among the entities of the information network, and it is hard to find the important information just through our intuitive feelings. So the research on the complex networks is particularly significant for the real system analysis. Recently, the information analysis has gained extremely wide attentions from researchers in many disciplines. And the information network analysis has become a hot research topic in data mining and information retrieval fields in the preceding decades. Particularly, the community detection is an important research content in the study of complex networks. Community detection is to detect useful and relatively stable community and valuable for network information collection and mining, information recommendations and predicting the evolution of networks.

However, most of the contemporary community detection analyses have a basic assumption: the type of the objects or links is unique [2]. In other words, the modeled networks are homogeneous networks containing the same type of objects and links. But most real systems contain multi-typed interacting components. For example, in the bibliographic database, like DBLP [2], papers are connected through authors, venues and terms. Although homogeneous network provides a convenient method for network analysis, it ignores the heterogeneity of objects and links or only considers one type of relations among one type of objects. Compared with widely-used homogeneous networks, the heterogeneous networks could effectively fuse more information and contain richer semantic information of nodes and links, and thus it could obtain more accurate results in community detection.

On the other hand, most of the contemporary community detection algorithms on heterogeneous networks only work on small dataset, and fail to consider the quick and parallel process on big data. So the time complexity of the traditional algorithm is so high that it can’t meet the requirement of the large-scale data. With the rapid expansion of the network size, the time complexity of the traditional algorithm has a trend of exponential growth. In order to detect the accurate community structure and illustrate the benefits of heterogeneous networks, we need to design algorithms on big networked data.

In this paper, considering the above problem, we propose and improve a method to detect the communities in heterogeneous networks and implement it based on spark graphs. Firstly, we simply introduced Rankles algorithm and optimize it according to the experimental environment. Then, we parallelize the algorithm on spark graphs to meet the requirement of the bigger networked data. Finally, we compare and discuss the experiment results using different master node and worker nodes, and show the different results with serial manner and parallel manner. Also, we present the community detection results of the improved Rankles algorithm and compare it with the original algorithm. Our research’s main contributions include:

We propose a method in community detection of heterogeneous networks instead of homogeneous networks.

We change the ranking function of Rankles, and substitute PageRank algorithm for simple ranking so as to achieve better results.

We make a research on parallel implementation in big networked data.

As graph computing based on spark get more attentions, we implement the algorithm on spark graphs to provide a reference for others subsequent research on community detection in heterogeneous networks.

The rest of our paper is organized as follows. Section 2 discuss some related works on community detection of heterogeneous networks. Section 3 introduced the background knowledge including the definition of heterogeneous network and Rankles algorithm. The details of improvement and optimization of Rankles and the parallelization process is described in Section 4. Section 5 presents the experiment results and comparison of different situations. Section 6 concludes our work and introduces some our future works.

RELATED WORKS

Community detection based on heterogeneous networks has drawn great attention in recent years. We summarize the related works for the past few years and group them into different categories: topic model method, data reconstruction method, dimensionality reduction method, ranking-based clustering and so on.

Text information plays an important role in many heterogeneous network studies. So many researchers make use of topic model to analysis the semantic information to increase the accuracy of community detection. Zhou et al. [3] propose CUT models for semantic community discovery in social networks, combining probabilistic modeling with community detection. Cha et al. [4] explore the application of the well-known LDA model and its existing variations for this task and propose two extensions to make LDA suitable for the social-network relationship graph. Considering the combination of semantic information and network structure, Mei et al. [5] and Cai et aal. [6] Propose Neoplasm model and LTM model respectively. Deng et al. [7] introduce a topic model with biased propagation to incorporate heterogeneous information network with topic modeling in a unified way. LSA-PTM [8] is introduced to identify clusters of multi-typed objects by propagating the topics obtained by LSA on the HIN via the links between different objects.

Because of the characteristics of multi-dimensional and multimode in the heterogenous, some researchers use data reconstruction method [9][10] and dimensionality reduction method [11][12][13] to convert it into relatively simple network type or reduce the dimension. Liu et al. [9] transform an original heterogeneous network into a bipartite network and perform community detection on it. Each node and edge or hyperedge in the original heterogeneous network is, respectively, mapped into a vertex node and a link node in the bipartite network. Wang et al. [11] propose a Bayesian probabilistic model, DBNMF model, for automatic detection of overlapping communities in temporal networks. Yang et al. [13] present BIGCLAM model, an overlapping community detection method that scales to large networks of millions of nodes and edges.

Recently, ranking-based clustering on heterogeneous information network has emerged, which shows its advantages on the mutual promotion of clustering and ranking. Sun et al. [14] propose RankClus algorithm to generate communities for a specified type of objects in a bi-type network based on the idea that the qualities of clustering and ranking are mutually enhanced. Furthermore, they also propose NetClus [15] to handle a network with the star-schema. Wang et al. [16] introduce Coleus to promote clustering and ranking performance by applying star schema network with self-loop to combine the heterogeneous and homogeneous information. But the above methods are aimed at some particular networks, so some researches [17][18] propose some general methods suitable for more kinds of network structures. Shi et al. [17] propose a general method HeProjI to do ranking based clustering in heterogeneous networks with arbitrary schema by projecting the network into a sequence of sub -

networks. Chen et al. [18] propose GPNRankClus model to achieve clustering and ranking simultaneously on a heterogeneous network with arbitrary schema.

Moreover, some researchers propose some methods from the other perspective in order to fuse the attribute information in heterogeneous network [19] [20] [21] [22]. Sun et al. [19] design a probabilistic model to cluster the objects of different types into a common hidden space, by using a user-specified set of attributes and the links from different relations. Boden et al. [20] propose the density-based clustering model TCSC for the detection of communities in heterogeneous networks that are densely connected in the network as well as in the attribute space.

PRELIMINARY

Heterogeneous Network

According to the type of nodes and edges, complex networks can be divided into homogeneous network and heterogeneous network. We define a heterogeneous network as follows.

Definition 1. Heterogeneous network [2]. An information network is defined as a directed graph $G=(V,E)$ with an object type mapping function $\varphi:V\rightarrow A$ and a link type mapping function $\psi:E\rightarrow R$. Each object $v\in V$ belongs to one particular object type in the object type set $A:\varphi(v)\in A$, and each link $e\in E$ belongs to a particular relation type in the relation type set $R:\psi(e)\in R$. If the types of objects $|A|>1$ or the types of relations $|R|>1$, it is a heterogeneous network.

Because the heterogeneous network has different types of objects or relationships, it has the characteristics of multi-mode and multi-dimensional. Thus the traditional methods of homogeneous network can't apply to the heterogeneous network and this make the research face new challenges.

Besides, unlike homogeneous networks, heterogeneous networks contain different kinds of network structures, as shown in Fig. 1, including bipartite network, k-partite network, star-schema network, multiple-hub network and so on.

As mentioned in Section 2, most of community detection methods are only applicable to specific network structure and it is hard to propose a method to detect community in all types of network structure.

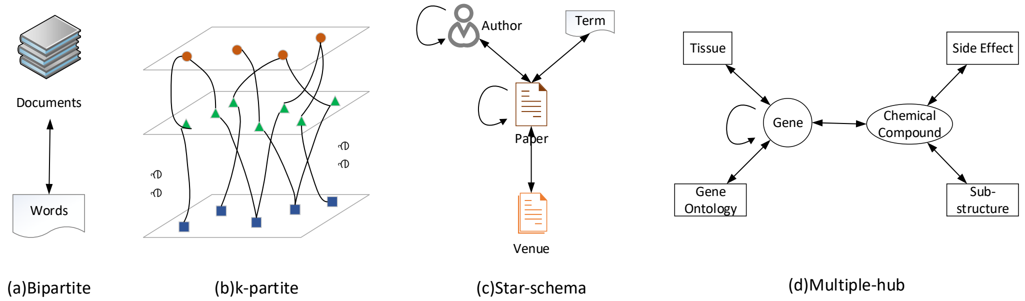

Figure 1. Network structure of heterogeneous networks.

GraphX

GraphX is a new component in Spark for graphs and graph-parallel computation. At a high level, GraphX extends the Spark RDD by introducing a new Graph abstraction: a directed multigraph with properties attached to each vertex and edge.

GraphX realized the distributed storage and graph procesing efficiently due to the inherent nature of RDD, and it could be applied to the large-scale graph computing. Compare to the MapReduce, graphx improve and optimize the performance of computing significantly. In addition, the optimization in storage of vertex and edge information make it close to or reach the performance of professional graph computing platform like GraphLab. So we choose graphx as our experimental platform.

ALGORITHM DESIGN

Improvement of Rankles

Rankles algorithm [14] could detect community of the bi-type network by combining the clustering with the ranking and enhance each other mutually. Rankles is a ranking-based method which let ranking score as the features for each community. That is to say, it divides the objects into different communities according to the ranking score. So ranking function plays an important role for the accuracy of the detection results.

Rankles is aimed at the bi-type heterogeneous network and proposes two ranking functions: simple ranking and authority ranking. Considering the efficiency of the algorithm, we choose the simple ranking to conduct an experiment and study.

We denote $G = \langle \{X \cup Y\}, W \rangle$ as the bi-type network, and X, Y is the object sets, W is the link matrix. W Can be written as:

$$W = \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{pmatrix} \tag{1}$$

Where the four blocks $W_{XX}, W_{XY}, W_{YX}, W_{YY}$ denote a sub-network of objects between types of the subscripts. The simple ranking generates the ranking score of the type X and type Y as follows:

$$\begin{cases} \vec{r}_X(x) = \dfrac{\sum_{j=1}^{n} W_{XY}(x, j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i, j)} \\ \vec{r}_Y(y) = \dfrac{\sum_{i=1}^{n} W_{XY}(i, y)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i, j)} \end{cases} \tag{2}$$

The time complexity of simple ranking is $\mathrm{O}(|\varepsilon|)$, where $|\varepsilon|$ the number of links is.

In view of the significant influence of ranking function on detection results, we make a research on the change of ranking function. PageRank could measure the importance of each vertex in a graph and is contained in the graphs. It is convenient to implement the experiment and will improve the efficiency of the algorithm. So we select PageRank as our ranking function, and in Section 5 we will show that it will improve the accuracy of the results without decreasing the efficiency.

In graphs, there are static and dynamic implementations of PageRank. Static PageRank runs for a fixed number of iterations, while dynamic PageRank runs until the ranks stop changing by more than a specified tolerance. We compute the PageRank of each object as shown in TABLE 1.

Parallelization design

Graph has a series of supersets in graph computing which contains three steps: local computation, communication and barrier synchronization. Graph completes the process of parallel computing among processors in one superset.

Rankles can be divided into three steps and each step must exchange the information. So we divide the algorithm into three supersets where each superset corresponds to one step of Rankles. On the other hand, we make use of PageRank to replace the original ranking function. The pseudo code of the parallel algorithm is shown in TABLE 2.

We should input the number of the communities, K, in advance and load the heterogeneous network. In the end, we'll obtain the K communities and the rank distribution for each community. First, we must divide objects into K nonempty communities. In superset one, we get the ranking score for each community using PageRank. In superset two, we get the centers $\vec{s}_{X_k}^{(t)}$ of each community based on the K dimensional vector $\vec{s}_{x_i}$ using EM algorithm. In superset three, calculate the distance between x and $X_k^{(t)}$ and assign each object x to the nearest community.

TABLE 1. IMPLEMENTATION OF PAGERANK.

```
//load package
import org.apache.spark.graphx.GraphLoader
//construct a graph
Val graph = GraphLoader.edgeListFile(sc, "the file directory of graph data")
//call PageRank function and return the vertex attribute
Val ranks = graph. PageRank( specified tolerance ).vertices
```

TABLE 2. PARALLEL PSEUDO CODE OF THE IMPROVEMENT ALGORITHM.

```
Algorithm: improvement of RankClus
Input: Bi-type Information Network $G = \langle X, Y; W \rangle$;
       Ranking Function $f$ = PageRank;
       Community Number $K$;
Output: $\{X_i\}_{i=1}^{K}$, $\vec{r}_{X|X_i}$, $\vec{r}_{Y|X_i}$;
//Initialization
t=0;
$X \rightarrow \{X_i^{(t)}\}_{i=1}^{K}$, get initial nonempty communities;
//Repeat Supersteps 1-3 until $< \varepsilon$ change or too many iterations
For (t = 1; t < iterNum && epsi > $\varepsilon$; t ++)
    //Superstep 1: Ranking for each community
    For i = 1 to K
        $G_i^{(t)} = \langle X_i^{(t)}, Y; W \rangle$, get subgraph from G;
        $(\vec{r}_{X|X_i}^{(t)}, \vec{r}_{Y|X_i}^{(t)}) = f(G_i^{(t)})$;
    End for
    //Superstep 2: Get new relation vectors for objects and communities
    Get $\vec{s}_{x_i}$ for each object $x_i$ using EM algorithm;
    For i = 1 to K
        $\vec{s}_{X_k}^{(t)}$ = get centers for community $X_k^{(t)}$;
    End for
    //Superstep 3: Adjust each object
    For each object x in X
        For i= 1 to K
            $D(x, X_k^{(t)})$ = distance between $x$ and $X_k^{(t)}$;
        End for
        Assign x to $X_{k_0}^{t+1}$, $k_0 = \arg\min_k D(x, X_k^{(t)})$;
    End for
End for
```

\When implement the above algorithm in graphs, we assign the heterogeneous network into K processors and each processor computes the data in parallel. The parallel computing is demonstrated in Fig. 2.

All processors communicate with each other in the process of communication and start the next superset when it reaches the process of barrier synchronization. We need to get the rank information from other processors in the communication process in superset one before calculate the centers of each community in superset two. Similarly, we must get the center vectors from other processors before calculate the distance $D(x, X_k^{(t)})$. Finally, after repeat the above three supersets we will get the optimal classification of community.

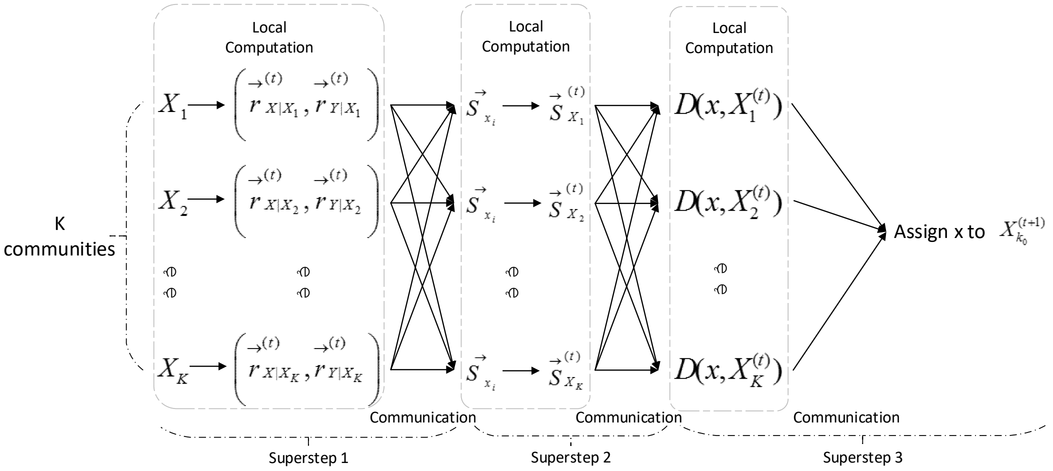

Figure 2. The process of parallel computing.

IMPLEMENTATION AND EXPERIMENT

Dataset and evaluation standard

In our experiment, we use DBLP [23] dataset to demonstrate our parallel algorithm. We downloads the raw data in XML file including publications, authors, conferences and journals. We extract a part as our experimental data which contains 28702 authors and 50 conferences or journals.

In the end, we obtain three txt files showing the author data, conference/journal data and the link relations respectively. The author-text file is shown as the format: ID + "\t" + author. The conference/journal-txt file is shown as the format: ID + "\t" + conference/journal. The link-txt file is shown as: object_ID1 + "\t" + object_ID2 + "\t" + frequency.

The conference/journal is our target object, so our purpose is to find the community structure of conferences/journals by ranking the authors. Considering the target object is a small number, we use a simple method to evaluate the accuracy of our experiment for convenience. The evaluation standard is defined as follows:

$$Accrucy\ Rate(AR) = 1 - \frac{Number\ of\ wrong\ classification}{Total\ number\ of\ conferences} \tag{3}$$

Environment

We establish our experimental environment on 6 computers where each has 4 cores (Intel i5), 4GB RAM and 2.8 GHz. Our system is Ubuntu 16.04 LTS and the version of JDK is jdk-8u131. The Hadoop version is 2.8.0 and the Spark version is 2.1.1. The SBT version is 0.13.15. Our programming language is Scala and its version is 2.11.8. We debug and run our code based on IntelliJ IDEA as our programming environment.

Experiment results

We conduct the experiment based on 6 computers where one computer serves as the master node and the others serve as worker node. We will compare the results between in series and in parallel. When there is no work node, that is, there only exists master node, the code is running in series. Of course, it is running in parallel when the work node is open. In addition, we also present the experiment result from different number of running work node. Next, we will demonstrate the results and compare from two aspects of accuracy and efficiency.

Accuracy Analysis: Fig. 3. Presents the results of accuracy from different running work nodes. As we can see, the accuracy of PageRank is superior to Simple Rank obviously in any case. For PageRank, AR is relatively low in series without work node. Because of the limitation of operating environment, the algorithm's iteration can't reach the best situation to get a better community detection results. When the code is running in parallel, AR increases significantly which also verify our analysis that the initial relatively low results low in series is due to the limitation of environment.

In addition, from the perspective of x axis, when the work node increases all results show a rising trend both in series and in parallel. But the rising trend is not significant when in parallel. Through the above analysis, we should know that the limitation of operating environment is much low in parallel, so the results won't have significant differences.

Efficiency Analysis: The efficiency analysis is shown in Fig. 4. The running time is different clearly between in series and in parallel. In series, the running time in both methods is about 1200s. And when we add one work node in experiment, the running time decreases significantly and the efficiency is almost doubled. Of course, as the work node adds to the running experiment, the running time decreases gradually in both methods.

On the other hand, the difference of efficiency is small between PageRank method and Simple Rank method. And the difference is far smaller with the increase of work node. As we can see, when there are one master node and four work node, the running time of the two methods is almost the same. So our proposed method could achieve a relatively high efficiency.

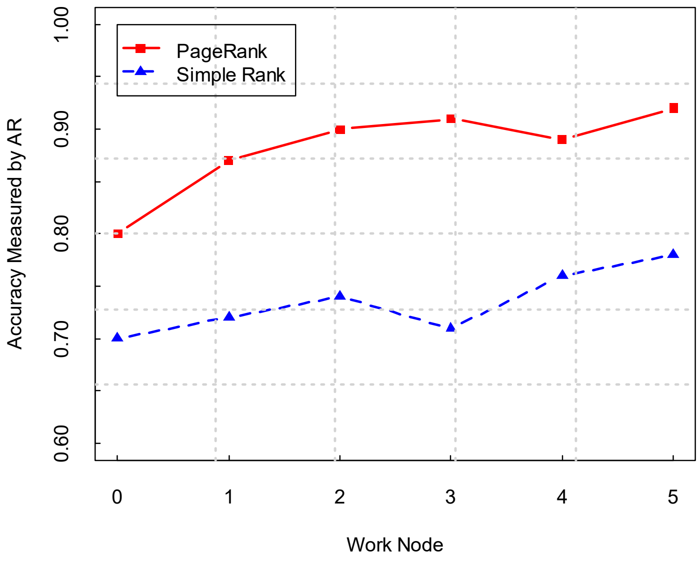

Figure 3. Accuracy Analysis.

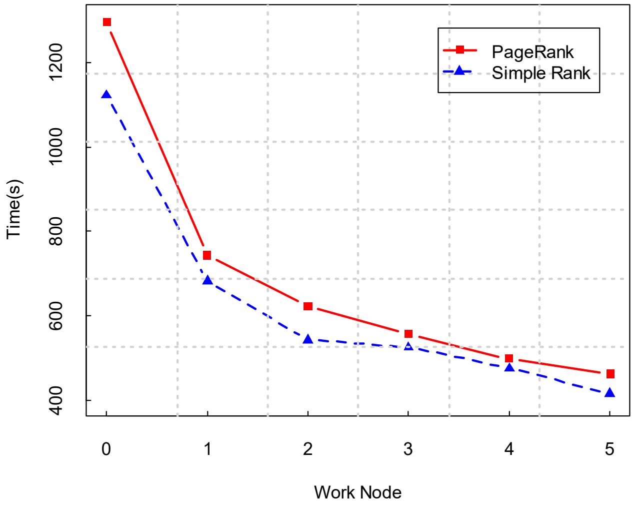

Figure 4. Efficiency Analysis.

From the analysis above, we can see that the improved RankClus algorithm based on graphx could reach the same efficiency as the original RankClus with simple rank function. Besides, our method increases the accuracy rate of community detection of bi-type heterogeneous network compared with the simple-rank-based RankClus.

CONCLUSION

In this paper, we propose a method applying to the community detection of bi-type heterogeneous network. We substitute PageRank for simple rank in RankClus, implement the method on graphx and analysis the results in parallel. Our experiment results confirmed that the parallel improved algorithm could reach a relatively high efficiency as with the simple-rank-based RankClus. Moreover, the accuracy is much better than the original RankClus algorithm.

In our experiment, we conduct the experiment based on spark graphx which could provide a reference for the following studies. But there are some limits and imperfection. For example, our dataset is relatively small, and we will collect more heterogeneous data to make further research in the future. Moreover, we will also focus on the authority-rank-based RankClus based on graphx and compare it with our method.

REFERENCES

1. Information on: http://www.199it.com/archives/437192.html.
2. Sun Y., Han J. Mining heterogeneous information networks: a structural analysis approach [J]. ACM SIGKDD Explorations Newsletter, 2013, 14(2): 20-28.
3. Zhou D., Manavoglu E., Li J., et al. Probabilistic models for discovering e-communities [C]//Proceedings of the 15th international conference on World Wide Web. ACM, 2006: 173-182
4. Cha Y., Cho J. Social-network analysis using topic models [C] //Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012: 565-574.
5. Mei Q., Cai D., Zhang D., et al. Topic modeling with network regularization [C]//Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 101-110.
6. Cai D, Wang X, He X. Probabilistic dyadic data analysis with local and global consistency [C]//Proceedings of the 26th annual international conference on machine learning. ACM, 2009: 105-112.
7. Deng H., Han J., Zhao B., et al. Probabilistic topic models with biased propagation on heterogeneous information networks [C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 1271-1279.
8. Wang Q., Peng Z., Jiang F., et al. LSA-PTM: a propagation-based topic model using latent semantic analysis on heterogeneous information networks [C]//International Conference on Web-Age Information Management. Springer, Berlin, Heidelberg, 2013: 13-24.
9. Liu W., Murata T., Liu X. Community Detection on Heterogeneous Networks [J]. Arid Environmental Monitoring, 2013.
10. Liu X., Liu W., Murata T., et al. A framework for community detection in heterogeneous multi-relational networks [J]. Advances in Complex Systems, 2014, 17(06): 1450018.
11. Wang W., Jiao P., He D., and et al. Autonomous overlapping community detection in temporal networks: A dynamic Bayesian nonnegative matrix factorization approach [J]. Knowledge-Based Systems, 2016, 110: 121-134.
12. Psorakis I., Roberts S., Ebden M., et al. Overlapping community detection using Bayesian non-negative matrix factorization [J]. Physical Review E, 2011, 83(6): 066114.
13. Yang J., Leskovec J. Overlapping community detection at scale: a nonnegative matrix factorization approach [C] //Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013: 587-596.
14. Sun Y., Han J., Zhao P., et al. Rankclus: integrating clustering with ranking for heterogeneous information network analysis [C]//Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009: 565-576.
15. Sun Y., Yu Y., Han J. Ranking-based clustering of heterogeneous information networks with star network schema [C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 797-806.
16. Wang R., Shi C., Philip S.Y., et al. Integrating clustering and ranking on hybrid heterogeneous information network [C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2013: 583-594.

17. Shi C., Wang R., Li Y., et al. Ranking-based clustering on general heterogeneous information networks by network projection [C] //Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014: 699-708.
18. Chen J., Dai W., Sun Y., et al. Clustering and ranking in heterogeneous information networks via gamma-poisson model [C]//Proceedings of the 2015 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2015: 424-432.
19. Sun Y., Aggarwal C.C., Han J. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes [J]. Proceedings of the VLDB Endowment, 2012, 5(5): 394-405.
20. Boden B., Ester M., Seidl T. Density-based subspace clustering in heterogeneous networks[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2014: 149-164.
21. Qi G.J., Aggarwal C.C., Huang T.S. On clustering heterogeneous social media objects with outlier links [C] //Proceedings of the fifth ACM international conference on Web search and data mining. ACM, 2012: 553-562.
22. Sun Y., Norick B., Han J, et al. Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks [J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2013, 7(3): 11.
23. Information on: http://www.dblp.org.-