Journal of

Software Engineering

ISSN 1819-4311

Academic
Journals Inc.

www.academicjournals.com

Journal of Software Engineering 9 (2): 242-253, 2015
ISSN 1819-4311 / DOI: 10.3923/jse.2015.242.253

An Improved k-Means Clustering Algorithm for the Community Discovery

Sun JiangYan
Modern Education Technology Center, Xi'an International University, Xi'an, 710077, China

ABSTRACT

Community discovery is, as the name implies, for founding useful community structure in the social network. So far, there are many mining the social network community algorithm and some of these algorithms have even got application in reality. However, it is important to note that most of the existed community discovery algorithms are only applicable to small and medium networks. This study proposes an improved k-means algorithm. First of all, the study uses two algorithms to obtain the initial clustering with high accuracy and adaptability. It can avoid the many processes after choosing the random initial value. Then this study uses the compression technology and B tree format to store community information which can effectively reduce the time complexity of matching nodes and space complexity of temporary data storage. Finally, this study proposes the timing and method of the trimming community so that it can get accurate community classification results in a small probability event.

Key words: Social network, community discovery, large data, k-means algorithm, similarity

INTRODUCTION

The community structure in social networks is closely related to the graphics division problem in computer science and hierarchical clustering problem in sociology (Le and Panchal, 2012). The k-means algorithm is proposed by James MacQueen (Covoes *et al.*, 2013; Peralta *et al.*, 2013). This algorithm is based on the idea of the greedy algorithm and the social network can be divided into two known community structures. Its basic idea is that the concept of a new gain function Q is introduced to divide social networks existing in the community structure. This function Q can represent the difference between the number of edges and two connecting edges in the structure of two community networks which can find the maximum value of the gain function to divide the community (Sipior *et al.*, 2004; Gonzalez *et al.*, 2014). Its specific strategy is that nodes of the community structure will be moved to other community structures or be switched in different community structures (Sah *et al.*, 2014; Liu *et al.*, 2014). Searching from the initial solution until better candidate solutions can not find from the current solution and stops.

The biggest drawback of the k-means algorithm is that the structure size of two community in the advanced community structure should be known. Otherwise, it is not possible to calculate the correct result. This limitation makes it is difficult to apply the network analysis algorithm in real networks. Because the prior knowledge of the community structure is often unknown. In addition, the algorithm is the same as all existing two sub-algorithms and it has the problem of how to know the number of community structure in the network and is hard to know when to stop the problem. This algorithm is generally applicable to small and medium sized network structure (Gloor *et al.*, 2008; Schilling and Fang, 2014). In this study, the improved clustering algorithm-k-means clustering algorithm can handle the community structure of the social network in the large data.

METHODOLOGY

The k-means clustering algorithm is improved under two standards. One is to make the k-means clustering algorithm applied to the community discovery problem and the standard should carefully be selected. The second is to minimize the time and space complexities for large data and ensure the accuracy of the algorithm. To complete the above two standards, references of some algorithms are essential. Before introducing the algorithm process, the brief idea of the algorithm should be proposed.

Firstly, the classic k-means algorithm is introduced. Next, for the community discovery problem, how to choose the relationship measure between nodes is proposed. The relationship measure is a high-dimensional relationship, so the next step is to introduce features of the relationship between high-dimensional. Finally, the prior idea is separately introduced in large data structures.

k-means algorithm: The k-means algorithm has been the most famous and common classification method in data mining algorithms and its principle is that k is known as the input parameter and n objects are divided into k clustering. The similarity is very high in a clustering and is too low out of a clustering. Where the clustering similarity is the mean measure about objects in the clustering and can be seen as the gravity center of the clustering.

The k-means algorithm works as follows:

- Randomly select k objects from the set and one object is a gravity center corresponding to a clustering
- Respectively calculate the similarity between the rest objects and k gravity centers and these objects are classified into the clustering with the lowest similarity
- According to clustering results, calculate k gravity centers
- Cluster all objects in the set according to new gravity center
- Repeat step 4 until the clustering result does not change
- Output results

The k-means algorithm can be used in the process of the community discovery which means n nodes can represent n objects. The Jaccard similarity between nodes can represent relationship measure between nodes and in the community the center point with the max similarity can replace the gravity center. Finally k nodes clustering can be obtained, i.e., k communities.

Characteristic of the dimension disaster in high dimension: High dimensional Euclidean space has some non-intuitive properties, i.e., the "dimension disaster". Non-Euclidean space has often abnormal condition. One respect of the "disaster" is that distances space among most points are equal in high dimensional. Another is that almost arbitrary between two vectors is approximately orthogonal:

- The distance distribution of high dimensional space

For a d dimensional Euclidean space, n points in a unit cube are randomly selected, that is to say, each point can be expressed as $[x_1, x_2,..., x_d]$, where each x_i are between 0 and 1. If d is 1, it is equivalent that points are randomly placed on a line with the length 1. At this time, the difference between point and point is very large. If d is too large, the Euclidean distance between the random point $[x_1, x_2,..., x_d]$ and $[y_1, y_2,..., y_d]$ is:

$$\sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \tag{1}$$

where, each x_i and y_i are uniformly random variables selected between 0 to 1. Sensibility, for any i, x_i-y_i may make a lot of results but d is too large. When all the x_i-y_i results are covered, the Euclidean distance between two points will become a fixed value. In fact, after a rigorous proof, when points are randomly selected in d dimensional space, almost the distance between points are close to the average distance.

If there is no close dotted pairs, it is fundamentally difficult to construct any clustering. There is almost no reason to gather dotted pairs to another clustering. Of course, the data can not be random. Even if the data is in high dimensional space, there exits useful clustering:

- The angle between vectors

Suppose that there are three random points A, B and C, where d is very large. Let A and C be the point [x_1, x_2,.., x_d] and [y_1, y_2,.., y_d], respectively and B is the origin of coordinate. The cosine of the angle ABC can be calculated by:

$$\frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2}\sqrt{\sum_{i=1}^{d} y_i^2}} \tag{2}$$

When d continues to grow, the denominator grows with d growth. The molecule is the sum of random values and it can be positive or negative. Therefore, the expected value of the molecule is 0. For a large d, the cosine of angles between any two vectors is almost close to 0 which means that the angle is approximately equal to 90 degree.

There is an important corollary in random vector orthogonal. Suppose that there are three random points A, B and C. The distance between A and B is d_1 and the distance between B and C is d_2. The distance between A and C is approximately equal to $\sqrt{d_1^2 + d_2^2}$.

Rain of thought in large data structure: If the k-means algorithm is directly used in the community network of the data structure, its time and space complexity is very difficult to be accepted. Some details of the classic k-means algorithm should be simplified:

- The method of node assignment in the classic k-means algorithm is to calculate the similarity between each node and k communities and the average time complexity of each node is O(k). The algorithm in this study lets the community be a balanced tree and the average time complexity is reduced to O(logk)
- In a community structure composed of many nodes, if a node is added to this structure and a new point is generated, the new center point is the one which is the closet to the old one. When calculating the center of a community, the new center can be calculated among a few points which are most similar to the old one without considering other nodes

- All data can not be stored in memory. Real exiting nodes are stored in the disk in this study and characteristics extracting from each community network are stored in the memory. All communities are stored in a balanced tree
- The accuracy of initial cluster is high. The algorithm proposed in this study will not repeat the process of point distributions. Only the split and merge of the community is considered

A good algorithm can take its internal structure and temporary data into consideration. Especially when dealing with large data structures, some simple improvements can significantly reduce the space complexity of the algorithm.

Representation of a single community: When assigning nodes to the community, the community will become larger. Most nodes of the community are only stored on disk which makes the distribution of nodes unavailable. To solve this problem, multiple features extracting from the community can be stored in memory and these features can not only represent all aspects of the community but also facilitate updating. Before listing these features, assume the node v_i is one point of the community. The ROWSUM(v_i) is the square sum of the similarity between v_i and all other points. Therefore, the ROWSUM(v_i) becomes higher and the node v_i is more likely to be the center of the community. Features of a community C_l in memory is shown in the following:

- The number of nodes n_l in the community
- The gravity center of the community. It is specifically defined as those nodes which has the largest similarity with others
- The ROWSUM value of the gravity center
- For a selected constant p, p nodes and its corresponding ROWSUM value have the most similarity to the community
- p nodes and its corresponding ROWSUM value have the least similarity to the community. p nodes can be used to judge whether two communities are close enough to merge

The value of p becomes larger and the accuracy of the algorithm is higher. In the large data structure, p is estimated as 0.1n%/k, where n is the number of all nodes in the net and k is the number of the initial clustering. Figure 1 is an example of a community.

In Fig. 1, its features are stored in the memory. n_l is 14 and nodes B, C and D has the best similarity to A and nodes E, F and G have the least similarity to A.

Community expressing tree: If all communities are organized into a tree, the number of tree nodes may be very large. This is because each node has to save more representations of the community which is specifically shown as follow:

- Internal nodes of the tree in the community save a sample of a sub-tree in all communities and pointers pointing to root nodes of the sub-tree. The size of these samples are fixed, so the number of sub-nodes of internal nodes is independent of depth. For an internal node, if its depth is lower, the number of the community in the sub-tree is high
- Nodes can represent a plurality of communities. The size of the community does not depend on community nodes but depend on the value of p

Based on the size of the memory and the community, the number of samples stored in internal nodes h and the number of the community g can be given.

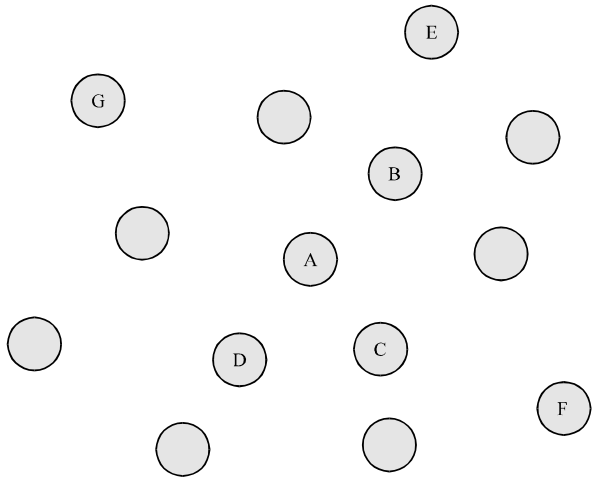

Fig. 1: A community

Two community expressing trees are shown in Fig. 2a and b. Figure 2a is a community expressing tree which has two levels. The tree has two leaf nodes which include C_1, C_2, C_3 and C_4, C_5, C_6, respectively. It also has an internal node which includes community centers of C_1, C_2, C_4. Gravity centers of C_1 and C_2 can represent the left leaf node and its pointer points to the left leaf node. The gravity center of C_4 can represent the right leaf node and its pointer points to the right leaf node.

Figure 2a is a community expressing tree which has two levels and it is also an balance tree. For all information of each layer, the information of the upper contains less information than the lower. Moreover, the correlation of some community stored in a tree is stronger than that of other community.

Whole process of the algorithm: Before introducing the whole algorithm, 3 processes are essentially introduced community representation tree initialization, nodes join, split and merge community.

Initialization of the community expressing tree: This study generates a tree S by using the tree T representing the community but not all trees are T. The specific process is as follows:

- Get the value from the bottom until a community is composed and input the sub-tree T' to the algorithm
- For each node of the tree T', calculate the the expression of the community and put the presentation of leaf nodes on the leaves the tree S. At present, a leaf node of S is filled with 1 or 2 initial clustering
- Merge nodes in S in accordance with its own information of the community. An internal node can be produced by every merging. h community information is randomly selected from the sub-nodes and the information is stored in the parent node
- General, if nodes in S are made fully in accordance with nodes in T', then the binary tree is obtained. However, in the large data structure, even if the community can save the, the binary tree may also not be stored entirely in memory. To reduce levels of the tree S, the binary tree

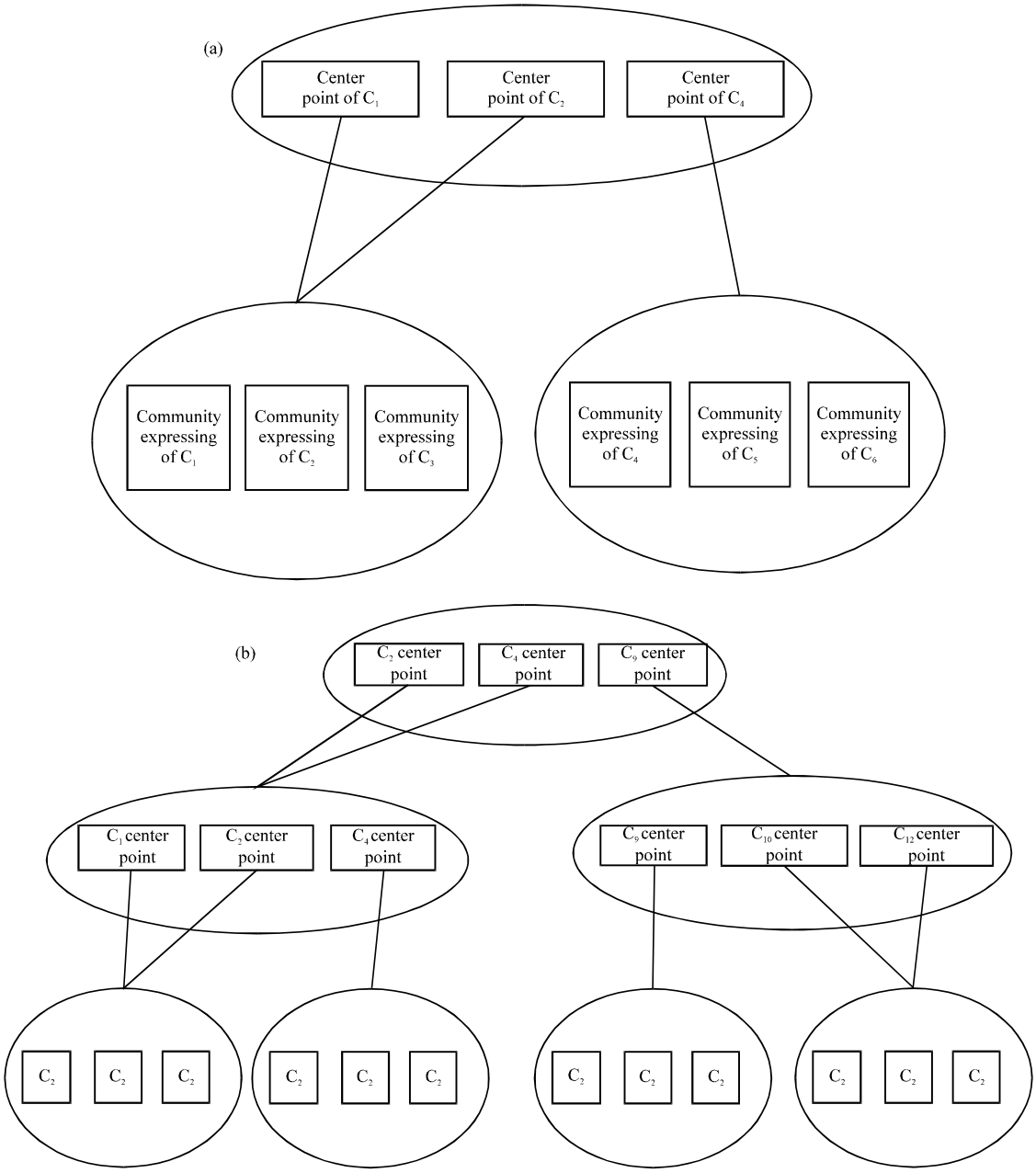

Fig. 2(a-b): A community expressing tree that has (a) Two levels and (b) Three levels

should have more branches of the B tree. On the other hand, the initial B tree can be unbalanced. The tree B should be balanced to ensure that each node is distributed in every branch of the tree

The whole initial process is shown in Fig. 3. Figure 3a is the result of transforming the sub-tree extracting form the tree T into the the tree T'. Figure 3b represents that the tree S is completely created in accordance with the father and son relationship of the tree T'. Figure 3c is the balanced tree B transformed from the binary tree.

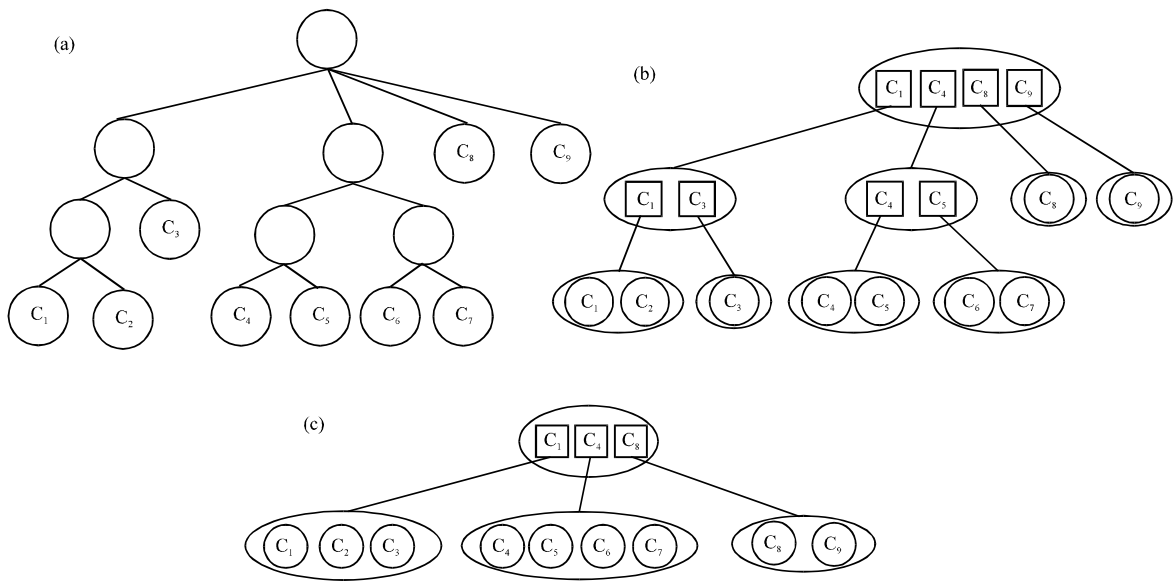

Fig. 3(a-c): An example for initializing a community expressing tree

Adding nodes: Now nodes reading from the disk should be inserted into the most similar community, specifically:

- Starting from the root node of the tree S, investigate center point samples of the community to find the most similar insertion point whose corresponding sub-node is the next inspected node
- Similarly, when an internal node in the tree S is traveled, center point samples should be investigated in the community. Insert the most similar center point into the community and determine which child node is the next one to visit
- Query the leaf node which includes a plurality of communities, where each community contains its corresponding features. The next step is to examine the community whose center point is the most similar to the insertion point
- After choosing a community, the insertion point is added to the community and the community representation is adjusted. Specifically speaking:
  - Let n_l plus 1
  - The quadratic sum between the insertion point i and the mentioned node j is added to each ROWSUM(v_j). These nodes j includes p nodes with the most similar to the center point and p nodes with the most dissimilar to the center point
  - Calculate similarity between the node i and the center point which is mainly used to determine whether i is the most similar node
  - If the insertion point is a part of this community expressing, ROWSUM must be calculated. Because there is no read from disk in the community all point, the ROWSUM value could not be accurately calculated. The following equation is used to make an estimate:

$$\text{ROWSUM}(v_i) = \text{ROWSUM}(v_{centre}) + n_l \times \text{Jaccard}^2(v_i, v_{centre}) \quad (3)$$

where, Jaccard(v_i, v_{centre}) is the similarity between the node i and the center point of the community. n_l and ROWSUM(v_{centre}) is the characteristic value before adding a new point

i. The effective of the equation can be proved by the "dimension disaster". A feature of the dimension disaster in high dimensional Euclidean space is that almost angles are right angles. If the node i, the center point center and another point j consists of an right angle, then according to the Pythagorean theorem, have:

$$\text{Jaccard}^2(v_i, v_j) = \text{Jaccard}^2(v_i, v_{centre}) + \text{Jaccard}^2(v_j, v_{centre}) \quad (4)$$

After calculating the ROWSUM value of the node i, the node i and its ROWSUM(v_i) will become a characteristic of the community which will replace another feature

- If the ROWSUM of the most similar node is less than ROWSUM(v_{centre}), the role of the node j should be swapped by the center point

Division and merging of the community: The following are two methods for the community structure adjustment.

Community division: The community center is one of the most important characteristic of the community. If multiple center points exits in a community, this community is too loose. It has to be split into some communities. Its characterization threshold P, in the data structure, can be expressed as n_{key}/k, where n_{key} is the number of critical points and k is the number of initial clustering.

Community merger: The community merger is generated by the community division. Considering the worst case of the community division, the current new tree are too large to fit in memory. This can only improve the threshold P to reduce the representation space and the two community should be merged.

To calculate the ROWSUM value in the initial community C_1 by using the following equation:

$$\text{ROWSUM}_C(v_i) = \text{ROWSUM}_{C1}(v_i) + n_{C2} \times (\text{Jaccard}^2(v_i, v_{C1}) + \text{Jaccard}^2(v_{C1}, v_{C2})) + \text{ROWSUM}_{C2}(v_{C2}) \quad (5)$$

where, ROWSUM_C, ROWSUM_{C1}, ROWSUM_{C2} are the calculation results of the community C, C_1, C_2 and n_2 is the number of nodes in the community C_2. v_{C1} and v_{C2} is the central point of the communities C_1 and C_2, respectively.

The above equation can be explained by the "dimension disaster". Suppose that the angle between the similarity of node v_i and v_{C1} and the similarity of node v_{C1} and v_{C2} is a right angle, the similarity between nodes v_i and v_{C2} can be calculated by the Pythagorean theorem:

$$\text{Jaccard}^2(v_i, v_{C2}) = \text{Jaccard}^2(v_i, v_{C1}) + \text{Jaccard}^2(v_{C1}, v_{C2}) \quad (6)$$

Suppose that the angle between the similarity of node v_i and v_{C2} and the similarity of node v_{C2} and v_j is a right angle, where j is a node of the community C_2 node. The similarity between nodes v_i and v_j can be calculated by the Pythagorean theorem:

$$\text{Jaccard}^2(v_i, v_j) = \text{Jaccard}^2(v_i, v_{C2}) + \text{Jaccard}^2(v_{C2}, v_j) \quad (7)$$

Finally, Eq. 6 and 7 plus $\text{ROWSUM}_{C1}(v_i)$ makes the Eq. 5.

Whole algorithm process: The above content mainly describes the implementation process of the algorithm. The whole algorithm process is described as follow:

- The initial clustering and a associated spanning tree T should firstly be obtained
- Initialize a community expressing tree S according to the structure of the tree T
- Read nodes from the disk and insert nodes to the best matching community in accordance with tree branch sequence of the tree S. For this community, recalculate the community representation
- Communities of the tree S should be split and merged in the necessary time

RESULTS AND DISCUSSION

In order to validate the performance of the proposed algorithm, this section will introduce the designed experiments so that the proposed evaluation method for the community structure can be proved. Firstly, the time consumption between the classical algorithm and the proposed algorithm is given; secondly, by using the visualization software to draw the community structure, the accurate and clear community structure can be compared with the known result to prove the algorithm; thirdly, in the case of limitation, the algorithm has the same reliability. The final results shows that, this algorithm can obtain better results in small or large data.

According to the use of the data set, the experimental data used in this study can be divided into 3 types.

Firstly, based on the benchmark data set, the correctness and effectiveness of the proposed can be analyzed. They are the dolphin social network and karate club network. The dolphin social network data set is composed of 62 nodes and 4 communities and the karate club network is composed of 34 nodes, 78 edges and 2 communities.

Secondly, a benchmark network data set is used to analyze time consumption. This data set is called the adolescent health network. This data set has 90118 nodes and 84 communities.

Finally, considering the large data, the computer only has 512 Mb memory; the real data set has 4847571 nodes and 68993773 edges-LiveJournal social network.

To adapt to the MATLAB, for each data set, the data should firstly be converted into the adjacency matrix A representing the social network topology. The dimensions of the matrix is the number of nodes in the network. If edges between nodes are originally connected, elements in A are 0 or 1. If original nodes have weights, then the the value A_{ij} in the A is the weights of nodes i and j.

Firstly, the dolphin social network and the karate club network data sets are inputted. By using the MATLAB, the result of community division can be imported into NodeXL as in Fig. 4a.

In Fig. 4b, the proposed algorithm can obtain the accurate result that the former data set is divided into 4 communities and the later data set is divided into 2 communities which is consistent with the real situation.

Next, the time consumption of this algorithm is discussed. The BGLL algorithm (Chaturvedi *et al.*, 2012) has low time complexity and the Nystrom algorithm has low space complexity. Table 1 shows the time consumption of 3 algorithms on 3 data sets (the dolphin social network, the karate club network, the adolescent health network).

Compared with the other two algorithms, the proposed algorithm consumes less time than the BGLL algorithm in the dolphin social network and adolescent health network. Especially in large data, the consumption of time keeps on the low level. This is at the cost of the accuracy by using the fuzzy computing and compression space.

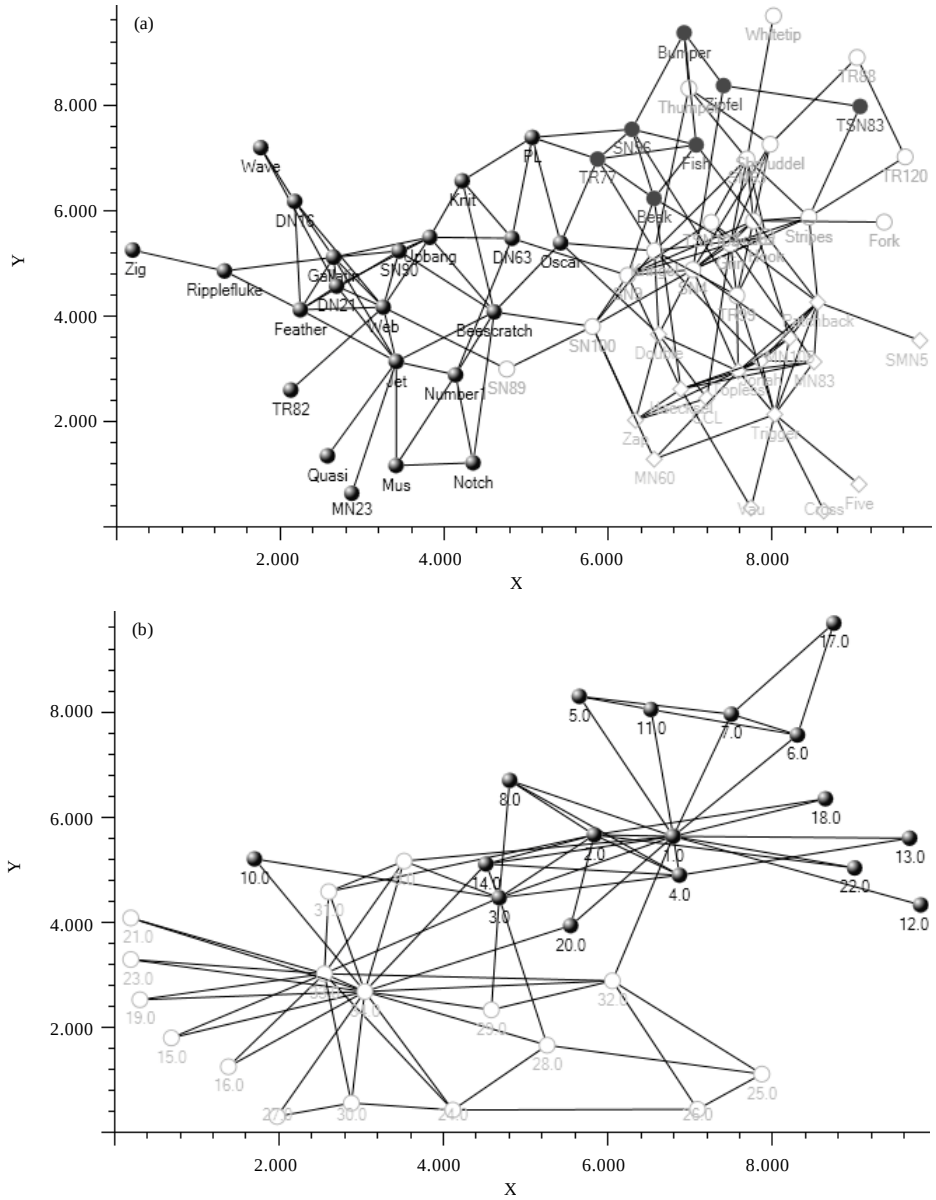

Fig. 4(a-b): Community discovery result of the (a) Dolphin social network and (b) Karate club social network

Table 1: Comparisons of time consumption

Algorithm	Karate club network	Dolphin social network	Adolescent health network
BGLL algorithm	0.01	0.03	261.83
Nystrom clustering algorithm	0.02	0.07	476.37
Proposed algorithm	0.02	0.06	159.41

Finally, to detect the algorithm's ability of adapting to large data process, the LiveJournal data set is handled in small memory of virtual machine. Before analyzing the proposed algorithm, the

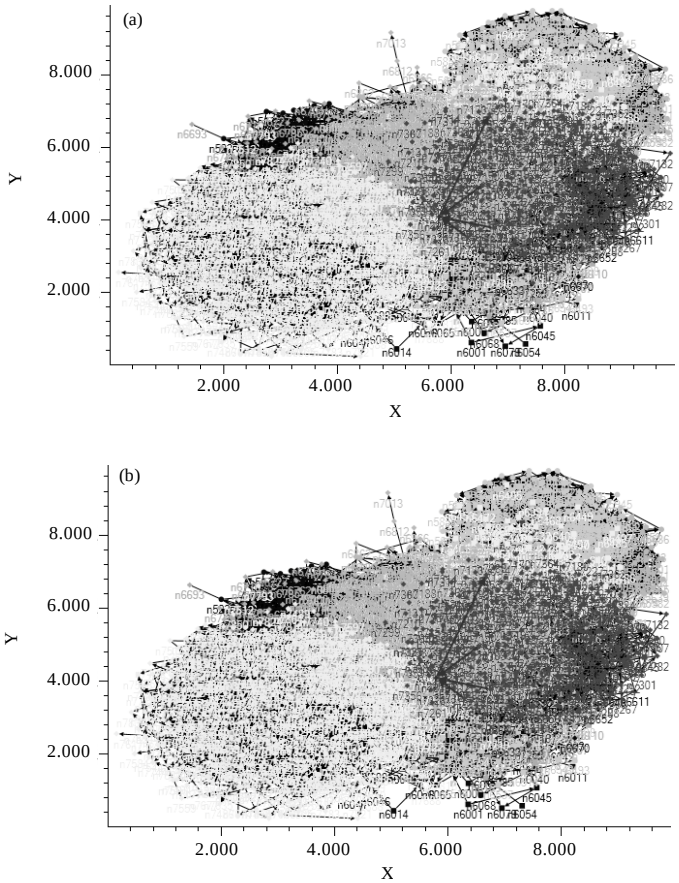

Fig. 5(a-b): Community discovery for LiveJournal social network data set (a) Nystrom algorithm and (b) BGLL algorithm

BGLL algorithm and the Nystrom spectral clustering algorithm have handled in the clustering processing. Results show that the BGLL algorithm can not run. The Nystrom algorithm can just work well before the initial sampling; after that, the memory thrashes which leads the algorithm to consuming too much time. While the proposed algorithm can get better results shown in Fig. 5a.

Finally, for the detection of this algorithm is the ability to adapt to large data processing, in small memory virtual machine in his social network LiveJournal data set. Before using this algorithm, we first tried to use the BGLL algorithm and the Nystrom spectral clustering algorithm for processing. Results BGLL algorithm can run; while the Nystrom spectral clustering algorithm works well after the initial sampling, in spectral clustering occurs when memory thrashing, the algorithm is time consuming. While using this algorithm can get good results, as shown in Fig. 5b.

CONCLUSION

This study has solved the community discovery problem in large data structure. By improving the classical k-means algorithm, the time and space complexity have been reduced to some extent in the premise of ensuring high accuracy. To compress the space, the feature of a community is

used in memory and the network structure is constructed by the balance B tree. To avoid reading communities into memory again and again, the new estimation value can be calculated by existing community features and the community split and merge process has been proposed. Finally, an experiment by using virtual and real data proves that the proposed algorithm has a good time and space complexity.

REFERENCES

Chaturvedi, P., M. Dhara and D. Arora, 2012. Community detection in complex network via BGLL algorithm. Int. J. Comput. Applic., 48: 32-42.

Covoes, T.F., E.R. Hruschka and J. Ghosh, 2013. A study of k-means-based algorithms for constrained clustering. Intell. Data Anal., 7: 485-505.

Gloor, P.A., M. Paasivaara, D. Schoder and P. Willems, 2008. Finding collaborative innovation networks through correlating performance with social network structure. Int. J. Prod. Res., 46: 1357-1371.

Gonzalez, G.R., D.P. Claro and R.W. Palmatier, 2014. Synergistic effects of relationship managers' social networks on sales performance. J. Market., 78: 76-94.

Le, Q. and J.H. Panchal, 2012. Analysis of the interdependent co-evolution of product structures and community structures using dependency modelling techniques. J. Eng. Des., 23: 807-828.

Liu, Y., J. Moser and S. Aviyente, 2014. Network community structure detection for directional neural networks inferred from multichannel multisubject EEG data. IEEE Trans. Biomed. Eng., 61: 1919-1930.

Peralta, B., P. Espinace and A. Soto, 2013. Enhancing k-means using class labels. Intell. Data Anal., 17: 1023-1039.

Sah, P., L.O. Singh, A. Clauset and S. Bansal, 2014. Exploring community structure in biological networks with random graphs. BMC Bioinform., Vol. 15. 10.1186/1471-2105-15-220

Schilling, M.A. and C. Fang, 2014. When hubs forget, lie and play favorites: Interpersonal network structure, information distortion and organizational learning. Strat. Manage. J., 35: 974-994.

Sipior, J., B.T. Ward, L. Volonino and J.Z. Marzec, 2004. A community initiative that diminished the digital divide. Commun. Assoc. Inform. Syst., 13: 29-56.