Refining Graph Partitioning for Social Network Clustering

Tieyun Qian[1] , Yang Yang[2], and Shuo Wang[2]

[1] State Key Laboratory of Software Engineering, Wuhan University
[2] International School of Software, Wuhan University
16 Luojiashan Road, Wuhan, Hubei, 430072, China
qty@whu.edu.cn, sherlockbourne@gmail.com,
wangshuokevin@gmail.com

Abstract. Graph partitioning is a traditional problem with many applications and a number of high-quality algorithms have been developed. Recently, demand for social network analysis arouses the new research interest on graph clustering. Social networks differ from conventional graphs in that they exhibit some key properties which are largely neglected in popular partitioning algorithms. In this paper, we propose a novel framework for finding clusters in real social networks. The framework consists of several key features. Firstly, we define a new metric which measures the small world strength between two vertices. Secondly, we design a strategy using this metric to greedily, yet effectively, refine existing partitioning algorithms for common objective functions. We conduct an extensive performance study. The empirical results clearly show that the proposed framework significantly improve the results of state-of-the-art methods.

Keywords: graph partitioning, small world property, network clustering.

1 Introduction

Graph partitioning is a fundamental problem, with applications to many areas such as parallel computing and VLSI layout. The goal of graph partitioning is to split the nodes in graph into several disjoint parts such that a predefined objective function, for example, ratio-cut, or normalized cut, is minimal. The optimal graph partitioning is NP-complete and a variety of heuristic or approximation algorithms have been proposed to solve this problem [1], [2], [3], [4], [5].

In recent years, with the increasing demand of social network analysis, cluster detection, or community discovery, arouses new research interest [6], [7], [8], [9]. Social networks show some unique properties different from those in conventional graphs. Among them, small-world property is a key feature of real life social networks. For example, new members in a club tend to be introduced to other members via their mutual friends. Topology in graphs derived from underlying meshes in parallel systems is to some extent regular and the degree distribution is comparatively uniform. In contrast, the neighborhood of a node in social networks is often densely

L. Chen, P. Triantafillou, and T. Suel (Eds.): WISE 2010, LNCS 6488, pp. 77–90, 2010.

connected and the entire network exhibits a strong characteristic of small-world. The small-world feature suggests the presence of strong local community structure and will surely affect the performance of clustering algorithm. Unfortunately, this key property is not considered in most of existing partitioning methods.

In this paper, we focus on the problem of graph clustering in social networks. In order to leverage the small-world property, we present a novel framework for enhancing the community quality obtained by popular partitioning algorithms. This framework is independent of the original approach and can be integrated with various types of clustering algorithms. In this paper, we choose three typical partitioning techniques, i.e., Metis, Fast Modularity, and Kernel K-Means, as the baseline methods for comparison.

The main contributions of the paper are:

- A new metric is defined to measure the small world strength between two vertices.
- A greedy yet effective strategy is presented to refine current partitioning methods.
- A system evaluation on several real world datasets show that the proposed approach can significantly improve the objective functions.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 describes the basic notions and notations. Section 4 defines a metric for measuring the small-world weight. Section 5 presents an algorithm for refining existing partitions of graph. Section 6 provides experimental results. Finally, Section 7 concludes the paper.

2 Related Work

A number of approaches have been developed for graph partitioning problem. The methods are categorized into three principle themes, i.e., geometric techniques, spectral methods, and multi-level methods. The geometric ones partition the graph based on coordinate information by putting vertices that are spatially near each other into one cluster. This kind of methods works only when coordinate information is available. The spectral methods are recently widely used in a number of fields including computer vision and speech segmentation [2], [5]. Spectral clustering relies on computing the first k eigenvectors corresponding to the Laplacian or some other matrix. Spectral approaches are in general computationally demanding and it is infeasible to compute the eigenvectors for large graphs. The third theme is based on multi-level schemes [3], [10], [11], which are considered as the state-of-the-art technique due to its high speed and quality. Multi-level partitioning methods consist of three main phases, i.e., coarsening phase, initial partitioning phase, and partition refinement phase. A sequence of successive approximations of the original graph is obtained in coarsening. And partitioning is performed on the smallest graph using recursive bisection methods. Finally, refinement on un-coarsened graphs is executed to gain the objective function by swapping pairs of nodes for Kernighan-Lin (K-L) method [12] or annotating each vertex along the partition for Fiduccia-Mattheyses (F-M) method [13]. As there is no strategy for guiding the search process, these two

refinements are very time-consuming, i.e. $\Theta(n^2 \log n)$ for K-L and $\Theta(n * \log n + e)$ for F-M method, where n and e are the number of nodes and number of edges in the graph. Recently, A. Abou-Rjeili and G. Karypis [10] examine the power-law property of real networks develop new strategies for graph clustering.

For most graph clustering methods, it is necessary to specify the number of clusters as input. However, people know little about the graph structure in advance. One solution to this problem is hierarchically clustering the groups into multi-level structure by merging the sub-clusters with the largest similarity, and let the user choose the best representation, as what agglomerative algorithms do [14]. Fast modularity algorithm [6] is another kind of hierarchically clustering algorithms by merging two clusters that increases a quality function named modularity Q. The Q value in each iteration is recorded and the largest one is chosen for deciding the best partitioning. This algorithm thus has the privilege that no input parameter or human interaction is required. The optimization of modularity is also NP-complete. Many modularity-based approaches have been presented along this line, with various optimization or search strategies such as simulated annealing [7], [15], [16], genetic algorithms [17], external optimization [18]. By embedding graphs in Euclidean space [8] or replacing the Laplacian matrix with the modularity matrix [19], spectral methods can also be used for modularity optimization. An intrinsic limitation of modularity based algorithms is that they are not appropriate for finding partitions whose sizes are diverse.

In summary, the problem of community detection is typically NP-hard and a lot of heuristics or approximation algorithms have been presented to optimize the objective function. In contradiction to the large amount of clustering methods, there is less research interest on partition refinement. Kernighan-Lin [12] and Fiduccia-Mattheyses [13] are two basic methods for refinement phase. The main drawbacks of these methods are the high time complexity. Thus they are not appropriate for enhancing large-scale networks. Most importantly, they are not developed specifically for social networks with unique properties. Though the approach presented in [10] utilizes the power-low feature of networks, it is only designed for multilevel partitioning, and cannot be added on the top of other partitioning methods like modularity and spectral clustering.

3 Problem Statement

The problem of graph clustering is to assign the vertices in a graph G into k disjoint groups such that a predefined quality function is optimized. Formally, given a graph $G=(V, E)$, where V is the vertex set and E is the edge set. The problem consists of dividing the vertices in G into k disjointed partitions $P_1 = (V_1, E_1), P_2 = (V_2, E_2), ..., P_k = (V_3, E_3)$ such that $V_i \cap V_j = \phi$, $\bigcup_{i=1}^{k} V_i = V$.

There have been a lot of metrics such as min-max cut [20] and ratio cut [21] proposed to measure the quality of graph clustering in literature. Among them, ratio cut, normalized cut [2], and modularity [6] are most commonly used.

Assuming a graph G is divided into k disjointed partitions $P_1, P_2, \ldots, P_k$. P_i is one of clusters and $G \backslash P_i$ is the rest of graph. The normalized cut is often used in image segmentation. It is defined as following:

$$NC = \sum_{i=1}^{k} \frac{|cut(P_i, G \setminus P_i)|}{d_{Pi}}, \tag{1}$$

Where d_{Pi} is the total degree of sub-graph P_i, $cut(P_i, G \setminus P_i)$ is the number of edges lying between P_i and the rest of graph.

The ratio cut is often used in the domain of circuit partitioning. It is defined as:

$$RC = \sum_{i=1}^{k} \frac{|cut(P_i, G \setminus P_i)|}{n_{Pi} n_{G \setminus P_i}}, \tag{2}$$

Where n_{Pi} and $n_{G \setminus Pi}$ is the sum of degrees of vertices in P_i and the rest of graph, respectively.

Both of these two metrics are proportional to the number of cut edges. The difference lies in the denominator. The smaller value of normalized cut or ratio cut, the higher quality of graph partition.

Recently, M. Newman presents a new function named modularity to judge the goodness of a graph partition. It is defines as:

$$Q = \sum_{i=1}^{k} \left(\frac{|e_{p_i}|}{|e_G|} - \left(\frac{|d_{p_i}|}{|2e_G|}\right)^2\right) \tag{3}$$

Where e_{Pi} is the number of internal edges in P_i and e_G the total number of edges in G. The larger modularity close to 1 means a good partition.

4 Measuring the Small World Weight between Two Nodes

Traditionally, for an un-weighted graph, the connection of two nodes is either 1 or 0. While for a weighted graph, the connection strength is represented as the edge weight. Formally, the edge weight of node pair (i, j) in G is defined as:

$$w_{ij} = \begin{cases} 1, \text{ if i and j are connected in an unweighted graph} \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

$$w_{ij} = \begin{cases} e_{ij}, \text{ if i and j are connected in a weighted graph} \\ 0, \text{ otherwise} \end{cases} \tag{5}$$

The adjacency information has been fully utilized in many existing graph partitioning methods to group nodes which are highly connected. However, in many applications, especially in social networks, the states of neighbors also have influence on the node pair. For example, if all the neighbors of one node *i* are connected with another node *j*, then node *i* and node *j* will have tighter connections, as if the neighbors are pulling them together. In contrast, the connection between node *i* and node *j* will become looser if all the neighbors of *i* are disconnected with *j*, as if the neighbors are pushing them away.

Figure 1 illustrates such phenomena.

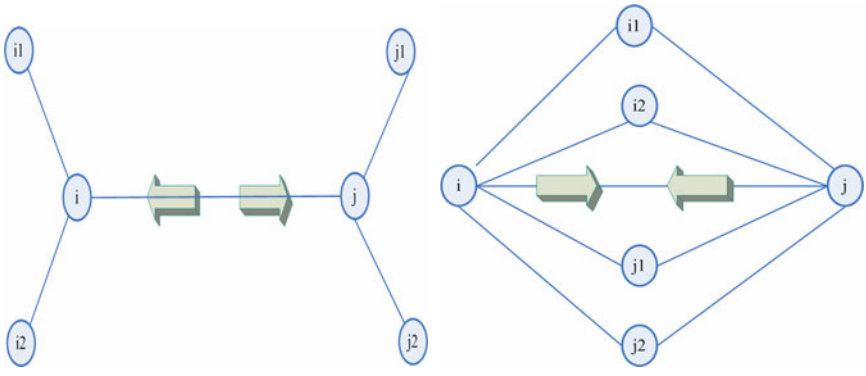


Fig. 1. Implications from nearby points

From Fig.1, we can see that besides the explicit relation between two nodes, there are also some implications from nearby points that will join or separate two vertices. Based on this observation, we propose a metric named small-world weight to measure such implications from neighborhood.

The small-world weight s_{ij} of node pair (i, j) is defined as follows:

$$s_{i->j} = \frac{\sum_{k \in \Gamma(i) \cap \Gamma(j)} w_{kj}}{\sum_{k \in \Gamma(i)} w_{ki}}, \quad s_{j->i} = \frac{\sum_{k \in \Gamma(i) \cap \Gamma(j)} w_{ki}}{\sum_{k \in \Gamma(j)} w_{kj}}, \tag{6}$$

$$s_{ij} = s_{i->j} + s_{j->i} \tag{7}$$

Where Γ_i and Γ_j is the neighbour hood of node i and node j respectively.

5 Algorithm for Refining Network Clustering

5.1 A Total Order for Generating the Sorted List of Node Pairs

A general description of the goal of graph clustering is that the edges are dense within a partition but between which they are sparse [6]. To this end, most existing techniques aim at reducing the cut edges between different clusters. As we have illustrated in previous section, this strategy may not work well for complex networks like social networks. In this paper, we present a three-dimensional criterion to define the connection strength for node pair. As the edge and small-world weights reflect the direct and indirect relations existing between nodes, we set them as the first and second

dimension accordingly. We also add the hub weight as the third dimension. The rationale is to promote the hub nodes. The hub value of node pair (i, j) is defined as follows:

$$h_{ij} = \max(d_i, d_j), \tag{8}$$

Where d_i and d_j is the degree of node i and node j respectively.

Combining these three dimensions together, we now define a total order on the node pairs. This is used in measuring the strength of connection between node pair.

Definition: Given two pairs of nodes (i_1, j_1) and (i_2, j_2), we say that $(i_1, j_1) \succ (i_2, j_2)$ or (i_1, j_1) has a higher precedence than (i_2, j_2) if:

1. the edge weight of (i_1, j_1) is greater than that of (i_2, j_2), or
2. the edge weights are the same, but the small-world weights of (i_1, j_1) is greater than that of (i_2, j_2), or
3. both the edge weights and the small-world weights of (i_1, j_1) and (i_2, j_2) are the same, but the hub value of (i_1, j_1) is greater than that of (i_2, j_2), or
4. all the edge weights, the small-world weights, and the hub values of (i_1, j_1) and (i_2, j_2) are the same, but node id i_1 is greater than that of i_2.

5.2 The Refinement Algorithm

Let L be the order list of node pairs and π the partitioning by running any graph clustering methods. The basic idea of the algorithm is to sequentially merge the node pair in L into one cluster if the following two conditions are satisfied: (1) they are currently separated, and (2) the merge operation will enhance the quality function. The steps for refining network clustering are shown in Algorithm 1.

Algorithm 1 Algorithm for refining network clustering (RENEC)
Input: a graph $G=(V,E)$, $\pi_0=\{P_1,P_2,...P_k\}$: the initial partition of G by a graph partitioning method, ω: a predefined objective quality function, L the sorted list of node pairs in G.
Output: π_i: the refined partition of G.
Method:

1. Calculate ω_0 of π_0
2. i = 0
3. while $L \neq \phi$ do
4. Fetch the first node pair (x, y) in L
5. Get the cluster labels l_x and l_y in ω_i
6. moved = false

7. if $l_x \neq l_y$
8. Calculate the temporary ω_{t1} by merging x to P_{l_y}
9. Calculate the temporary ω_{t2} by merging y to P_{l_x}
10. if $\omega_{t1} > \omega$ or $\omega_{t2} > \omega$
11. $\omega_{i+1} = \max(\omega_{t1}, \omega_{t2})$
12. $\pi_{i+1} = \pi_{t1}$ if $\omega_{t1} > \omega_{t2}$, otherwise $\pi_{i+1} = \pi_{t2}$
13. moved = true
14. if moved = false
15. $\omega_{i+1} = \omega_i$
16. $\pi_{i+1} = \pi_i$
17. $L = L - \{(x, y)\}$
18. i = i+1
19. Return π_i

In Algorithm 1, line 1 calculates the quality function value of initial partition. Line 2 initiates iteration variable. Lines 4-5 fetch the first node pair (x, y) in L and get their cluster labels. Line 6 initiates a boolean variable to judge whether a moving operation actually happens. If the two vertices of the node pair are in separated clusters, then calculate the temporary quality function values ω_{t1} and ω_{t2} by merging x to y's cluster or merging y to x's cluster in lines 8-9. If either of these changes makes improvement, then lines 11-12 save the merging operation with larger value of objective functions, and line 13 sets the boolean variable. If the two nodes are in the same cluster, or the moving operation can not make any enhancement, lines 15-16 remain current partitions. Line 17 removes the handled pair from the ordered list L. Line 18 update iteration variable. This procedure repeats until all pairs have been tested. Line 19 returns the final refined partitioning.

The objective function ω in our algorithm can be of any quality metrics for graph partitioning. Specifically, in this paper, we adopt three kinds of functions, i.e., normalized cut, ratio cut, and modularity. In addition, the initial partition of G can be made by *any* clustering methods such as spectral clustering and multilevel partitioning.

6 Experimental Results

6.1 Experimental Datasets

6.1.1 Datasets

We have conducted experiments on 4 typical social network data sets. The first one is the subset of Cora citation dataset [22] in the field of machine learning research. The second one is the CA-hepth collaboration network [23]. The third one is the Enron email communication network [24]. The forth one is the Epinions online interaction network [25].

Table 1 shows the number of nodes and edges of the datasets after pre-processing.

Table 1. Summary of data sets

dataset	|nodes|	|edges|
Cora-7ml	27891	65755
CA-hepth	9877	25975
Enron	36692	183831
Epinions	75879	405740

6.1.2 The Network Properties

Small-world networks have a small diameter and a high clustering coefficient. These properties as well as the density of the studied datasets are shown in Table 2. The Enron network has the smallest diameter and largest clustering coefficient, exhibiting a significant small-world feature. The Cora-7ml network has the smallest value of clustering coefficient and the lowest density. The clustering coefficient of Epinions network is also relatively small. However, the density of this graph is very large. The diameter of CA-hepth network is large.

Table 2. Properties of the studied networks

dataset	diameter	clustering coefficient	density
Cora-7ml	7.413	0.1261	2.36
CA-hepth	8.866	0.4714	2.63
Enron	5.978	0.4970	5.01
Epinions	6.329	0.1378	5.35

6.2 Improvements over FM Method

Fast modularity (FM) approach [6] is one of the most popular algorithms for community discovery. It does not require any hypothesis on the number of clusters. The objective in FM is modularity Q.

We first run FM algorithm on three datasets, and then apply the proposed RENEC algorithm. Figure 2 shows the improvements of Q values. The enhancements of modularity on Cora-7ml, Ca-hepth and Enron datasets are not very significant. Modularity value Q is often used to test whether a particular division is meaningful [6]. Q values greater than about 0.3 appear to show significant community structure. Initial Q values for these three datasets indicate that all these networks have already split quite cleanly into groups, and hence smaller enhancements can be made. The proposed method improves Q value for Epinions dataset from 0.358 to 0.409, reaching an improvement ratio of 14.25%.

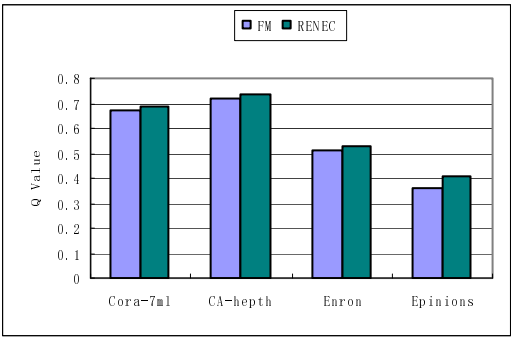


Fig. 2. Improvements of *Q Value* of RENEC over FM

6.3 Improvements over Metis

We choose Metis algorithm [3] as the second baseline clustering method, because it provides high quality partitions better than those produces by spectral partitioning algorithms, also because it is fast and requires less memory. The objective of Metis is to minimize the edgecut. Hence we set Normalized Cut (NC) (RC) or Ratio Cut as the objective functions to verify whether they can be further enhanced by our RENEC algorithm.

It is necessary to pre-assign the number of clusters, say k. A commonly used setting for this parameter is the square root of the number of nodes. We will investigate the sensitivity to the parameter k later.

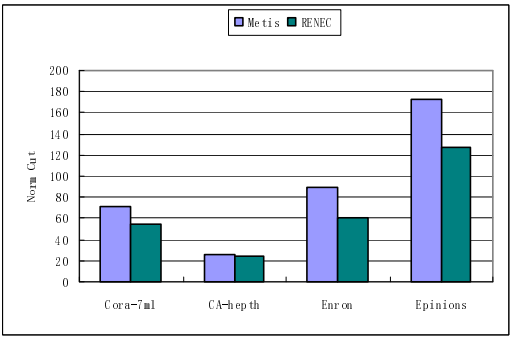


Fig. 3. Decrease of *Normalized Cut* of RENEC over Metis

Figure 3 and Figure 4 show the improvements of our RENEC over Metis in terms of Normalized Cut value and Ratio Cut value, respectively. Note that for these two evaluation metrics, the smaller, the better.

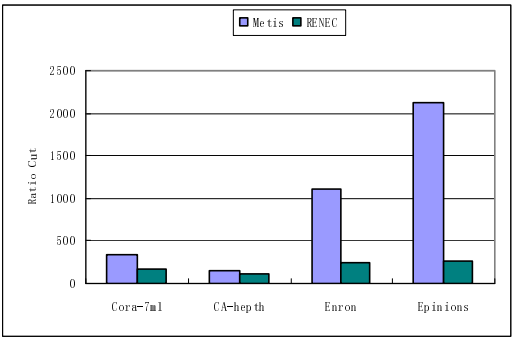


Fig. 4. Decrease of *Ratio Cut* of RENEC over Metis

The reduction of normalized cut on Enron is the biggest. It dramatically decreases from 89.02 to 61.27, a 31.17% improvement. The result on Epinions is also very attractive, reaching 26.92%. Despite of the diverse absolute ratio cut values for Metis, namely, 2120.75 for Epinions and 142.79 on Ca-hepth graph, the reductions of ratio cut are very similar to those of normalized cut. The reduction of ratio cut on Enron and Epinion reaches 77.40% and 87.43, respectively. The decrease on Ca-hepth and Cora-7ml datasets in these figures seems less significant. Please note that this is because of the different scale. For example, in Figure 4, the ratio cut for Cora-7ml dataset drops from 336.94 to 173.76, reaching a change ratio of 48.43%.

From Fig. 3 and Fig. 4, we can clearly see that, in all 4 networks, the proposed RENEC approach can significantly decrease the normalized cut and ratio cut obtained by Metis algorithm.

6.4 Improvements over Kernel K-Means Method

The third baseline clustering method is Kernel K-Means approach [26], which performs well for the task of linked-based clustering. The normalized cut and ratio cut are adopted as evaluation metrics.

Figure 5 and Figure 6 show the improvements of our RENEC over K-K-Means in terms of Normalized Cut value and Ratio Cut value, respectively. Again, the proposed RENEC approach can significantly decrease the objective functions obtained by K-K-Means algorithm. Take the Epinions online interaction network as an example, the decrease of Ratio Cut of RENEC over K-K-Means is very impressive. It drops from 5687.42 to 241.37, reaching a change ratio of 95.76%.

We notice that Ca-hepth dataset has the least reduction of normalized cut and ratio cut. The main reason can be that both Metis and K-K-Means methods work well for this kind of network, given the fact that the original function values on Ca-hepth are already the smallest among 4 graphs.

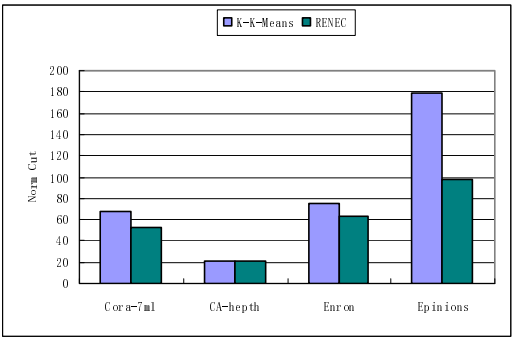


Fig. 5. Decrease of *Normalized Cut* of RENEC over K-K-Means

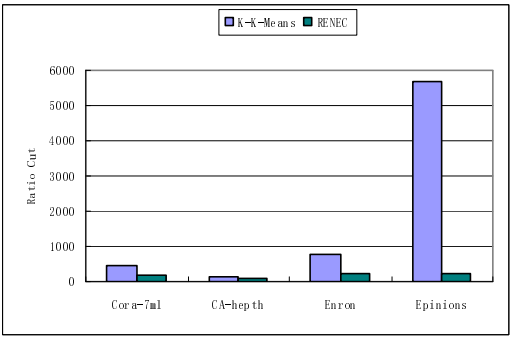
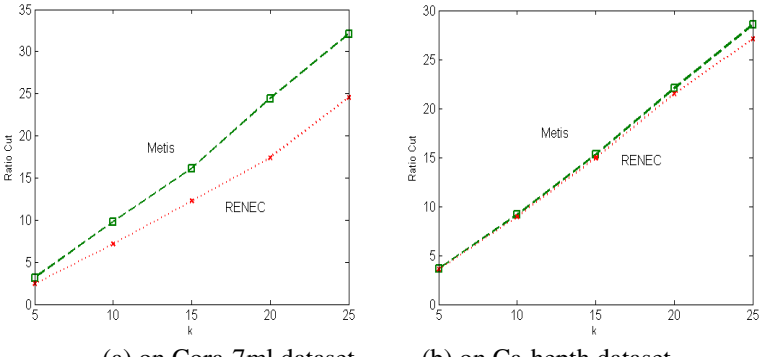


Fig. 6. Decrease of *Ratio Cut* of RENEC over K-K-Means

6.5 Sensitivity to the Number of Clusters

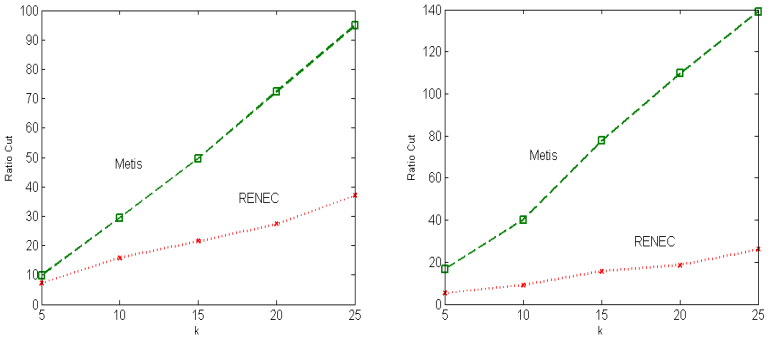
In this subsection, we evaluate the effect of the parameter k on the performance of the refinement approach by varying the input number of clusters. We set k to 5, 10, 15, 20, and 25, respectively.



(a) on Cora-7ml dataset (b) on Ca-hepth dataset

Fig. 7. Sensitivity of *Ratio Cut* to the number of clusters on Cora-7ml and Ca-hepth datasets

From the results in Figure 7 and Figure 8, both the values of ratio cut and the effects of the refinement algorithm increase with the growth of the parameter k on 4 data sets. This is reasonable as larger k will bring larger number of cut edges, as well as more chance for changing the destinations of the nodes.



(a) on Enron dataset (b) on Epinion dataset

Fig. 8. Sensitivity of *Ratio Cut* to the number of clusters on Enron and Epinion datasets

The benefit of using the proposed framework is clearly demonstrated in this experiment. Note that we also observe the similar improvements in terms of normalized cut and modularity on all other datasets. We do not present these results here only due to space limits.

6.6 Time Efficiency

Table 3 shows the timing results for the proposed method. It can be seen that the algorithm is very fast. All experiments were performed on a PC with Core 2.4 GHz CPU and 4G byte of memory running the Windows Vista Operating system. Please note that due to the memory limit, we implement the experiments using the data structure of adjacent list. If other data structure such as matrix is used, the time efficiency for our method can be further improved.

Table 3. Time efficiency of the proposed RENEC approach

dataset	k	CPU Execution Time in Seconds	
		time for sorting	time for refining
Cora-7ml	167	17	2
CA-hepth	99	3	0.2
Enron	192	118	12
Epinion	275	608	40

7 Conclusion

The small-world feature is one of the most important properties for social networks. Existing algorithms for graph partitioning do not take this key property into

consideration. This paper presents a refinement approach for adjusting the current graph partitioning methods. By incorporating the small-world and hub weight with the edge weight, we successfully define a three dimensional total order on edges. The ordered list is further used to merge the node pairs into one cluster which are currently separated by the original partitioning method. The extensive evaluation on several real-world social network datasets demonstrates the effectiveness of the proposed approach over other state-of-the-art techniques.

Acknowledgments. This research was supported in part by the 111 Project (B07037), the NSFC Project (60873007), the NSF of Hubei Province (2008CDB340), Specialized Research Fund for the Doctoral Program of Higher Education, China (20090141120050).

References

1. Bui, T., Jones, C.: A heuristic for reducing fill in sparse matrix factorization. In: 6th SIAM Conf. Parallel Processing for Scientific Computing, pp. 445–452 (1993)
2. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. In: Proc. of CVPR, pp. 731–737 (1997)
3. Karypis, G., Kumar, V.: A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing 20(1), 359–392 (1998)
4. Leighton, F., Rao, S.: Multi-commodity max-flow min-cut theorems and their use in designing approximation algorithms. J. ACM 46(6), 787–832 (1999)
5. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 14, pp. 849–856. MIT Press, Cambridge (2001)
6. Newman, M.: Fast algorithm for detecting community structure in networks. Phys. Rev. E 69 art. (066133) (2004)
7. Guimerà, R., Sales-Pardo, M., Amaral, L.A.N.: Modularity from fluctuations in random graphs and complex networks. Phys. Rev. E 70(2), 025101 (R) (2004)
8. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proc. of SIAM International Conference on Data Mining, pp. 76–84 (2005)
9. Tang, L., Wang, X., Liu, H.: Uncovering Groups via Heterogeneous Interaction Analysis. In: Proc. of ICDM, pp. 503–512 (2009)
10. Abou-rjeili, A., Karypis, G.: Multilevel Algorithms for Partitioning Power-Law Graphs. Technical Report, TR 05-034 (2005)
11. Hauck, S., Borriello, G.: An evaluation of bipartitioning technique. In: Proc. Chapel Hill Conference on Advanced Research in VLSI (1995)
12. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal 49(2), 291–307 (1970)
13. Fiduccia, C.M., Mattheyses, R.M.: A linear time heuristic for improving network partitions. In: Proc. 19th IEEE Design Automation Conference, pp. 175–181 (1982)
14. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer, Berlin (2001)
15. Massen, C.P., Doye, J.P.K.: Identifying communities within energy landscapes. Phys. Rev. E 71(4), 46101 (2005)
16. Medus, A., Acuña, G., Dorso, C.O.: Detection of community structures in networks via global optimization. Physica A 358, 593–604 (2005)

17. Tasgin, M., Herdagdelen, A., Bingol, H.: Community detection in complex networks using genetic algorithms, eprint arXiv: 0711.0491
18. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Phys. Rev. E 72(2), 27104 (2005)
19. Newman, M.E.J.: From the cover: Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA 103, 8577–8582 (2006)
20. Ding, C.H.Q., He, X., et al.: A min-max cut algorithm for graph partitioning and data clustering. In: Proc. of ICDM, pp. 107–114 (2001)
21. Wei, Y.-C., Cheng, C.-K.: Towards efficient hierarchical designs by ratio cut partitioning. In: Proc. of Intl. Conf. on Computer Aided Design, pp. 298–301. Institute of Electrical and Electronics Engineers, New York (1989)
22. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the Construction of Internet Portals with Machine Learning. Information Retrieval Journal 3, 127–163 (2000)
23. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph Evolution: densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD) 1(1) (2007)
24. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2005)
25. Richardson, M., Agrawal, R., Domingos, P.: Trust Management for the Semantic Web. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 351–368. Springer, Heidelberg (2003)
26. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: a multilevel approach. IEEE. Transactions on PAMI 29(11), 1944–1957 (2007)