Modular Community Detection in Networks

Wenye Li
Macao Polytechnic Institute
Macao SAR, China
wyli@ipm.edu.mo

Dale Schuurmans
University of Alberta
Edmonton, Canada
dale@cs.ualberta.ca

Abstract

Network community detection—the problem of dividing a network of interest into clusters for intelligent analysis—has recently attracted significant attention in diverse fields of research. To discover intrinsic community structure a quantitative measure called *modularity* has been widely adopted as an optimization objective. Unfortunately, modularity is inherently NP-hard to optimize and approximate solutions must be sought if tractability is to be ensured. In practice, a spectral relaxation method is most often adopted, after which a community partition is recovered from relaxed fractional values by a rounding process. In this paper, we propose an *iterative rounding* strategy for identifying the partition decisions that is coupled with a fast *constrained power method* that sequentially achieves tighter spectral relaxations. Extensive evaluation with this coupled relaxation-rounding method demonstrates consistent and sometimes dramatic improvements in the modularity of the communities discovered.

1 Introduction

Many important systems can be represented as networks, with entities represented by vertices and relationships represented by edges. Prominent examples include the world wide web, social networks, biological networks, communication networks, etc. [Easley and Kleinberg, 2010]. Research on networks has attracted significant recent interest, particularly in computing sciences and artificial intelligence, in response to the rapid increase in size and availability of real world networks and the practical needs to analyze them.

When analyzing such networks, an important question has often been “How many communities are there and what are the memberships?”. Community (i.e. cluster) structure seems to be inherent in real-world networks: vertices tend to cluster in groups where vertex connections within the same group are dense, while the connections are sparser between vertices from different groups. The ability to find and analyze such groups has proved invaluable in understanding network structure.

Computationally, the quality of a partition obtained depends on the quality of the objective function being used [Leskovec *et al.*, 2010]. Recently, the *modularity* function, Q, which measures the quality of a particular grouping of vertices in a network, has been widely accepted. Girvan and Newman [2002] have shown across a variety of simulated and real-world networks that larger Q values are correlated with better graph vertex groupings.

Unfortunately, maximizing Q is fundamentally difficult, hence heuristic approximation methods have been proposed for locally optimizing it. Among them, a spectral method proposed [Newman, 2006] has attracted broad attention. After relaxation, the method computes a decision vector where each element corresponds to the partition assignment of a vertex. To recover a hard partition from such a relaxed solution it has been standard practice to round each element individually based simply on their sign. Although simple, this *conventional rounding* strategy has achieved good empirical results and has been deployed extensively in the analysis of real-world networks and other graph partition applications.

In this paper we propose an *iterative rounding* strategy for recovering the final decisions. Unlike conventional rounding, which purely operates on the individual signs, we take the magnitude of each element into consideration in a sequential manner. That is, in successive rounds only a portion of elements with large magnitudes are rounded to hard decisions. The remaining elements are then re-optimized in the next iteration by solving a *residual* problem. The solution to the residual problem is again partially rounded into decisions, and so on. At the core of our proposal is a new *constrained power method* that achieves fast computation of the residual problem. This sequential approach more tightly approximates the global modularity objective by interleaving partial rounding with tighter spectral relaxation of the successive residual problems. Through extensive evaluations, the iterative rounding method reports significant and consistent improvement over the conventional approach.

2 Preliminaries

Modularity is the standard objective function used in network cluster analysis. It quantifies the quality of a given division of a network into communities. Good divisions, which have high modularity values, are those with dense edge connections between the vertices within a community but sparse connections between vertices in different communities.

Consider an undirected graph $G = (V, E)$ where $V = \{v_1, v_2, \cdots, v_n\}$ is a set of vertices and E is a set of edges between vertex pairs. Let w_{ij} be an element of the adjacency matrix W of the network, which gives the number of edges between vertices v_i and v_j. We further denote $d_i = \sum_j w_{ij}$ as the degree of v_i and $m = \frac{1}{2}\sum_i d_i$ as the total edge number.

For a candidate partition of the vertices into clusters, the *modularity* is defined to be the portion of the edge connections within the same cluster minus the expected portion if the connections were distributed randomly. Assuming the degree d_i associated with each vertex v_i is preserved, under uniform random selection the expected number of edges between two vertices v_i and v_j is $\frac{d_i d_j}{2m}$. Thus the observed number minus the expected number is $w_{ij} - \frac{d_i d_j}{2m}$. Summing over all pairs of vertices within the same group, the modularity, denoted by Q, is given by

$$Q = \frac{1}{2m}\sum_{ij}\left[w_{ij} - \frac{d_i d_j}{2m}\right]\delta(c_i, c_j)$$

where c_i is the group to which vertex v_i belongs, and δ is the Kronecker delta function.

The value of Q lies in the range $[-1, 1]$. It is positive when the observed connections within the same group exceed the expected number under random connections. Given a larger than expected portion of connections, one can reasonably infer the presence of an underlying cluster structure. Thus, the cluster structure can be searched precisely by checking the network divisions that have large modularity values.

An equivalent formulation is often used. Define s_{ir} to be 1 if vertex v_i belongs to group r and 0 otherwise. Then $\delta(c_i, c_j) = \sum_r s_{ir}s_{jr}$ and hence

$$Q = \frac{1}{2m}\sum_{ij}\sum_{r}\left[w_{ij} - \frac{d_i d_j}{2m}\right]s_{ir}s_{jr} = \frac{1}{2m}tr\left(S^T BS\right)$$

where *tr* denotes the trace of a matrix, S is the matrix having elements s_{ir}, and B is the *modularity matrix* having elements

$$b_{ij} = w_{ij} - \frac{d_i d_j}{2m}.$$

All rows and columns of the modularity matrix sum to zero, which means that the modularity of an undivided graph is always zero.

Unlike most statistical clustering models or graph partition techniques, which require a prior setting of partition numbers or group sizes [Jain *et al.*, 1999; Shi and Malik, 2000; Ng *et al.*, 2002], the modularity score determines the partition number and the group size automatically without manual intervention. This measure also allows the possibility that no good division of a network exists, corresponding to the case that the modularity value is zero (and cannot be increased by further division of vertices).

3 Spectral Modularity Maximization

Maximizing Q is NP-hard [Brandes *et al.*, 2006], therefore researchers have sought approximate solutions. In practice, a spectral relaxation method is widely used, that obtains reasonable empirical results in both optimization accuracy and computation time [Newman, 2006].

3.1 Two-Way Partitions

To first understand the spectral method, consider a simple case where the graph is divided into two groups. One defines $s_i = \pm 1$ to indicate the group membership of v_i, yielding

$$Q = \frac{1}{4m}\sum_{ij} b_{ij}s_i s_j = \frac{1}{4m}s^T Bs$$

where s is the column vector with elements s_i.

The vector s can be expressed as a linear combination of the normalized eigenvectors u_i of the modularity matrix B, so that $s = \sum_{i=1}^n a_i u_i$ with $a_i = u_i^T s$. Then one obtains

$$Q = \frac{1}{4m}\sum_i a_i u_i^T B \sum_j a_j u_j = \frac{1}{4m}\sum_{i=1}^{n}\left(u_i^T s\right)^2 \lambda_i,$$

where λ_i is the eigenvalue of B corresponding to the eigenvector u_i.

Assume that the eigenvalues are labeled non-increasingly, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. To maximize Q, the assignment vector s needs to concentrate as much weight as possible in the terms involving the leading (largest algebraic) eigenvalues, which, if s were unconstrained, could be achieved by setting s proportional to the leading eigenvector u_1. But with "±1" constraints, s cannot be chosen freely, which makes the optimization difficult.

Fortunately there is a convenient approximation available. Ignoring the inconvenient fact that it is not possible to make s perfectly parallel to u_1, one simply divides the vertices into two groups according to the signs of each element of u_1. Although this approximation is straightforward, it has often been found to give reasonable results in practice.

3.2 Multi-Way Partitions

The simple two-way partition method can be extended to multi-way partition method recursively. That is, using successive two-way partitions that divide the graph into subgraphs, the process can be continued on each subgraph until no further increases in Q can be found.

Formally, for each subgraph $G' = (V', E')$ with n' vertices we define an $n' \times n'$ subgraph modularity matrix B' with elements

$$b'_{ij} = w_{ij} - \frac{d_i d_j}{2m} - \delta(i, j)\left(d'_i - d_i\frac{m'}{m}\right),$$

where $d'_i = \sum_{j:v_j \in V'} w_{ij}$ and $m' = \frac{1}{2}\sum_{i:v_i \in V'} d_i$. The subgraph modularity is given by $Q' = \frac{1}{4m}s'^T B' s'$, where s' is a column vector with n' elements. Maximizing Q' with respect to s' gives the further contribution to the modularity Q obtained by subdividing the subgraph. When $G' = G$, B' reduces to B since $d'_i \to d_i$ and $m' \to m$ in that case.

The division process on each subgraph is halted when there exists no division that further increases the graph modularity; that is, no division that yields a positive value for Q'. This happens when the modularity matrix B' has no positive eigenvalues, hence the leading eigenvalue provides a simple check for terminating the division process.

3.3 Combination with Exchange Heuristics

In practice, the spectral method is often used in conjunction with exchange heuristics. In this approach, one uses spectral partitioning to obtain an initial broad division of the graph into subgraphs, then refines this division by moving vertices between groups using the Kernighan-Lin algorithm [Lin and Kernighan, 1973].

Given two groups of vertices, the refinement proceeds as follows. Successively find the vertex that, when moved to the other group, obtains the largest increase in Q, or the smallest decrease if no increase exists. Repeatedly make such moves, but ensuring that each vertex is moved only once. When all vertices have been moved, search all intermediate states to find the division that obtained the greatest Q. Starting again from this state, repeat the exchange process, until no further improvement is possible for Q.

As reported in [Newman, 2006], this combination gives excellent results on many open networks and has become a standard baseline when comparing community detection algorithms.

3.4 Computational Issues

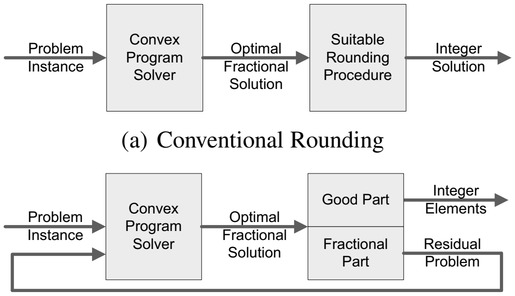
The modularity matrix B has special structure that can be exploited to efficiently compute the leading eigenvector via the power method. The power method approximates the dominant eigenvalue (the eigenvalue with the largest magnitude) and its eigenvector by iteratively multiplying a given matrix with an initial vector. For example, starting from a random vector v^0, the power method iteratively refines v by matrix-vector multiplication and renormalization $v^{i+1} = \frac{Bv^i}{\|Bv^i\|}$, and approximates the dominant eigenvector of B efficiently.

For modularity matrices, although the leading eigenvector might not be dominant, one can still apply the power method by using a simple trick. Without loss of generality assume the eigenvalues of B satisfy $\lambda_1 > \lambda_2 \geq \cdots \geq 0 \geq \cdots \geq \lambda_{n-1} > \lambda_n$. Using the power method, first compute B's dominant eigenvalue. If the eigenvalue is positive, it is λ_1, and its eigenvector is precisely the leading eigenvector u_1 we seek. If the eigenvalue is negative, it is λ_n, the most negative eigenvalue. In this case, shift the matrix $B + \frac{|\lambda_n|}{2} I$, where I is an $n \times n$ identity matrix. The shifted matrix has eigenvalues $\lambda_1 + \frac{|\lambda_n|}{2}, \lambda_2 + \frac{|\lambda_n|}{2}, \cdots, \frac{\lambda_n}{2}$ but the same eigenvectors as B. Applying the power method to this new matrix returns $\lambda_1 + \frac{|\lambda_n|}{2}$, with the desired eigenvector u_1.

With trivial modifications, the power method can also be used to calculate the leading eigenvector for the subgraph modularity matrix B'. We omit the details here.

4 Iterative Rounding for Community Detection

We now present our main proposal for improving the modularity of the communities discovered in a network.



(a) Conventional Rounding

(b) Iterative Rounding

Figure 1: Conventional Rounding and Iterative Rounding.

4.1 Conventional Rounding vs Iterative Rounding

In the spectral approach, given the computed eigenvector, a vertex partition is usually recovered by simple rounding

$$s_i = \begin{cases} +1 & u_{1i} > 0 \\ -1 & otherwise \end{cases}$$

where u_{1i} denotes the i-th element of the leading eigenvector u_1. We refer to this strategy as *conventional rounding*.

Note that the conventional rounding strategy is based on the signs of the eigenvector elements, regardless of their magnitudes (absolute values). However, the elements with different magnitudes contribute differently to Q, and therefore affect the confidence in the rounding decisions. For example, if u_{1i} has a large magnitude, then s_i will have a significant influence on $u_1^T s$ in the objective, and we would be more confident in inferring its rounded value. However, if the magnitude is small, s_i's contribution to the objective is less evident, and one would be less confident to make the rounding decision. In this latter case, we would like to postpone the rounding decision to a later phase.

Based on the idea, we propose a successive rounding method that only rounds variables with top magnitudes. That is, unlike conventional rounding that makes the entire partition decision in a single batch, we propose to recover more accurate community structure incrementally, using a strategy we will refer to as *iterative rounding*.

The two rounding schemes are illustrated in Figure 1. In iterative rounding, we first find an approximate solution to the original problem. Then, given a relaxed result, we do not turn all the elements into decisions like conventional rounding, but instead only round those with large magnitudes. Then we proceed to the next iteration by studying the *residual* problem, which often has a structure similar to the original but with fewer elements. This process is repeated until no variables are left un-rounded.

4.2 Constrained Power Method

To explain how the residual problems are efficiently solved in the context of iterative rounding, consider an illustration of the approach. In the first iteration, we have the same problem, $\max s^T Bs$, as the conventional spectral method. We simply use the power method and get the leading eigenvector. Then, rather than deploy conventional rounding, we only round the elements with largest magnitudes into decisions.

After the first iteration, we are left a residual problem to solve. Now suppose $s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$ where s_1 denotes the rounded elements that are held fixed, and s_2 contains the unrounded elements yet to be set. We can re-write the objective as the maximization of

$$\begin{aligned} & \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}^T \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \\ = \quad & s_2^T B_{22} s_2 + 2 s_2^T B_{21} s_1 + s_1^T B_{11} s_1 \end{aligned}$$

where B_{11}, B_{12}, B_{21} and B_{22} are four sub-matrices of B. Equivalently we have a residual problem to maximize

$$L = s_2^T B_{22} s_2 + 2 s_2^T B_{21} s_1$$

with respect to s_2, subject to the unit length constraint.

We can apply the gradient-based method for the maximization by iteratively updating s_2 along the gradient direction and renormalizing it, with which the convergence is guaranteed.

We also find a procedure we call *constrained power method* that gives excellent results. In gradient-based update s_2 converges when the gradient ∇L is parallel to the current estimate of s_2, or $\nabla L = 2\lambda s_2$ where λ is a scalar number. Then we have

$$s_2 = \frac{B_{22} s_2 + B_{21} s_1}{\lambda}.$$

We can force λ to be positive. Since s_2 is of unit length, it must hold that $\|B_{22} s_2 + B_{21} s_1\| = \lambda$. Thus we reach a simple update rule for s_2:

$$s_2^{i+1} = \frac{B_{22} s_2^i + B_{21} s_1}{\left\|B_{22} s_2^i + B_{21} s_1\right\|}.$$

The update occurs in a similar manner as in the power method. When $B_{21} s_1 = 0$, the constrained power method reduces exactly to the power method.

Starting with the previous fractional result as s_2^0, the constrained power method often converges quickly in practice. Given the updated relaxed solution produced by the constrained power method, we again only round those elements in s_2 that have large magnitude. After this partial rounding, we are left another residual problem sharing the same structure. The constrained power method and iterative rounding are applied successively for each residual problem. The process is terminated once s_2 has no elements, hence all vertex grouping decisions have been made.

It is also possible to apply the projected power method developed in [Xu *et al.*, 2009] for the optimization problem in each iteration, which exhibits similar performance as the constrained power method on community detections. We omit the details here.

4.3 Complexity Analysis

To analyze the time complexity for iterative rounding in spectral partitions, we first borrow some results on the power method from [Newman, 2006]. For a sparse network with n vertices and $m \leq kn$ edges where k is a constant, the power method effectively requires $O(n)$ matrix-vector multiplications to converge, where each multiplication requires $O(n)$ floating point operations. In total, the power method requires $O\left(n^2\right)$ time to calculate the leading eigenvector of the modularity matrix.

Similarly to the power method, the constrained power method has a complexity of $O\left(n^2\right)$ for a problem with n variables. Furthermore, suppose after each iteration an ϵ ($0 < \epsilon \leq 1$) fraction of the variables are rounded into integer decisions. Then in the subsequent round the residual problem becomes one with $n(1-\epsilon)$ variables, and the constrained power method would therefore require $O\left(n^2(1-\epsilon)^2\right)$ operations to converge. Repeating this argument, the complexity of iterative rounding becomes:

$$\begin{aligned} & n^2 + n^2(1-\epsilon)^2 + n^2(1-\epsilon)^4 + n^2(1-\epsilon)^6 + \cdots \\ = \quad & \frac{1}{2\epsilon - \epsilon^2} n^2 \\ \leq \quad & \frac{1}{\epsilon} n^2 \end{aligned}$$

and we come to the following observation.

For a network with n vertices, the complexity of iterative rounding is $O\left(\frac{1}{\epsilon} n^2\right)$ where ϵ is the fraction of variables to round in each iteration.

Comparing with the complexity of $O\left(n^2\right)$ for conventional rounding in network community detection, we can see the difference is up to a factor $\frac{1}{\epsilon}$, which is usually a constant value in practice. Suppose in each iteration $\frac{1}{4}$ of the variables are rounded, then the run time of iterative rounding would be around 4 times the run time of the conventional rounding method. Although the estimate is not theoretically strict, as we will see in our evaluations, it fits practice well.

5 Evaluation

We compared the proposed iterative rounding strategy to the conventional spectral method. In a series of experiments we observed consistent and sometimes very large improvements. In particular,

- for two-way partitions (cf. Section 3.1), iterative rounding demonstrates significantly improved Q values;
- for multi-way partitions (cf. Section 3.2), iterative rounding demonstrates significantly improved Q values;
- for partitions refined by exchange heuristics (cf. Section 3.3), iterative rounding demonstrates significantly improved Q values;
- iterative rounding has reasonable computation overhead comparing with conventional rounding.

These evaluations were conducted on all fourteen networks contained in a standard benchmark collection. [1] These benchmark networks cover a variety of application areas and are briefly described in Table 1.

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

Table 1: Networks used in the evaluations and their sources. The vertex numbers are listed in brackets.

Network	Source
karate	friendships of 34 members in a karate club
(34)	[Zachary, 1977]
dolphins	frequent associations of 62 dolphins
(62)	[Lusseau *et al.*, 2003]
lesmis	character interactions from *Les Misérables*
(77)	[Knuth, 1993]
polbooks	co-purchase of politics books at *Amazon.com*
(105)	(www.orgnet.com)
adjnoun	adjacency of adjs and nouns in *David Copperfield*
(112)	[Newman, 2006]
football	US college football games during Fall 2000
(115)	[Girvan and Newman, 2002]
celegans	neural network of C. Elegans
(297)	[Watts and Strogatz, 1998]
polblogs	hyperlinks of US politics web logs
(1224)	[Adamic and Glance, 2005]
netsci	co-authorship on network theory and experiment
(1461)	[Newman, 2006]
power	topology of US Western States Power Grid
(4941)	[Watts and Strogatz, 1998]
hepth	co-authorship on preprints of *High-Energy Theory*
(7610)	[Newman, 2001]
astroph	co-authorship on preprints of *Astrophysics*
(16046)	[Newman, 2001]
condmat	co-authorship on preprints of *Condensed Matter*
(16264)	[Newman, 2001]
internet	snapshot of Internet in level of autonomous systems
(22963)	(www.routeviews.org)

5.1 Two-Way Partitions

In this case, we divided each network into two partitions and checked the Q value obtained by two rounding methods respectively. The results are listed in Table 2. On all fourteen networks, iterative rounding demonstrates improved Q values over conventional rounding.

For small networks, two rounding methods reported comparable Q values and the improvement from iterative rounding is not large. On the smallest "karate" network with 34 vertices, the improvement is only slight, from 0.371 to 0.372. However, on larger networks the improvements are much more significant. One example is on "hepth" network with $7,610$ vertices, where the conventional rounding result is 0.034 while the iterative rounding achieves a modularity score of 0.455.

5.2 Multi-Way Partitions

Next, we investigated multi-way partitioning, wherein two-way partitions are repeated until no further increase of Q is possible. These results are listed in Table 2, where it can be seen that improvements achieved by iterative rounding remain consistent. On all networks, iterative rounding obtained better Q values than conventional rounding.

Similarly to the two-way case, the improvement is particularly significant on large networks. On "karate" network, the increase is only from 0.393 to 0.417; while on the larger "hepth" network, it becomes 0.739 to 0.829; and on the largest "internet" network the improvement is from 0.419 to 0.620.

Table 2: Q values by conventional rounding (CR) and iterative rounding (IR). Each item has three values $(A/B/C)$. A is for two-way partitions, B is for multi-way partitions, and C is for multi-way partitions refined by exchange heuristics. For networks with over $10,000$ vertices, the refined results were not obtained on our computer.

Networks	CR	IR
karate	.371/.393/.420	.372/.417/.420
dolphins	.390/.491/.519	.403/.526/.526
lesmis	.361/.532/.550	.381/.551/.560
polbooks	.445/.467/.521	.457/.523/.527
adjnoun	.191/.243/.308	.214/.298/.308
football	.376/.493/.599	.400/.601/.605
celegan	.261/.332/.400	.313/.381/.401
polblogs	.424/.424/.425	.426/.426/.426
netsci	.131/.671/.908	.496/.953/.954
power	.062/.898/.924	.491/.933/.934
hepth	.034/.739/.812	.455/.829/.839
astroph	.195/.586/	.417/.725/
condmat	.210/.677/	.453/.823/
internet	.301/.419/	.370/.620/

5.3 With Exchange Heuristics

As discussed in Section 3.3, the spectral partition method is often used in conjunction with exchange heuristics, which provides further refinement for the community partitions discovered. In our experiments, we also compared the Q values achieved by the two rounding methods after this fine-tuning.

The refined Q values are listed in Table 2. On "karate" network, both methods successfully detected the partition structure with $Q = 0.420$ which is known to be optimal via mathematical programming [Agarwal and Kempe, 2008]. On "adjnoun" network, the two methods obtained the same result after refinement. On all the other nine networks where the exchange heuristic was applicable on a conventional computer, iterative rounding achieved improved Q values to conventional rounding.

An interesting observation is that iterative rounding benefits less from the exchange heuristic. For conventional rounding, the exchange heuristic helps to increase Q significantly on most networks. However for iterative rounding, the increase is not as evident. On the other hand, on eight out of eleven networks, the iterative rounding multi-way results alone (without refinement) are better than the conventional rounding results with refinement, further exhibiting the effectiveness of iterative rounding.

5.4 Run Time Comparison

We recorded the run time of conventional rounding and iterative rounding on networks with more than $1,000$ vertices. Both rounding methods were implemented in Matlab and run on an Intel Xeon workstation with 32G RAM. For iterative rounding, we set $\epsilon = \frac{1}{4}$, meaning that $\frac{1}{4}$ of the variables were rounded in each iteration.

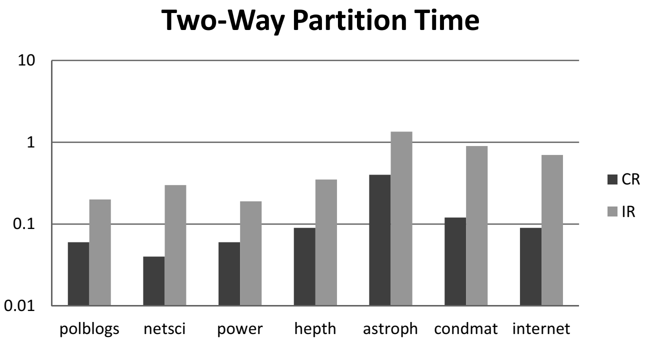


Figure 2: Run time (in seconds) of two-way partitions by conventional rounding (CR) and iterative rounding (IR) on networks with over $1,000$ vertices.

The two-way partition time is listed in Figure 2. The multiway partition time is not listed because the two rounding methods typically produce partitions with different numbers and sizes which makes the results not directly comparable. From these results, we can see that iterative rounding brings reasonable computation overhead, given the additional time required to solve the residual problems. It is approximately 3 to 7 times slower than conventional rounding in the experiment, consistent with our complexity analysis in Section 4.3.

6 Conclusion

This paper studies the community detection problem in networks. For this problem, a spectral relaxation algorithm is widely used in combination with a standard rounding strategy. Our work focuses on improving the results of these methods by a sequential rounding approach. This rounding strategy has achieved significantly improved results in empirical studies.

The study of rounding strategies is difficult. Not much work has been investigated [Fleischer *et al.*, 2006]. In practice, however, the strategy can be extremely important, especially for relaxation-based methods. As we have seen in Section 5, it sometimes governs the success of an algorithm. Thus more investigations along this line deserve our further attention. We expect such results to be applicable to a variety of related spectral partition problems, such as normalized cut.

References

[Adamic and Glance, 2005] L.A. Adamic and N. Glance. The political blogosphere and the 2004 U.S. election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[Agarwal and Kempe, 2008] G. Agarwal and D. Kempe. Modularity-maxmizing graph communities via mathematical programming. *European Physical Journal B*, 66, 2008.

[Brandes *et al.*, 2006] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity – NP-completeness and beyond, 2006.

[Easley and Kleinberg, 2010] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[Fleischer *et al.*, 2006] L. Fleischer, K. Jain, and D.P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci.*, 72(5):838–867, 2006.

[Girvan and Newman, 2002] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of National Academy of Sciences*, 99:7821–7826, 2002.

[Jain *et al.*, 1999] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[Knuth, 1993] D. Knuth. *The Stanford graphbase: a platform for combinatorial algorithms*. ACM Press, 1993.

[Leskovec *et al.*, 2010] J. Leskovec, K.J. Lang, and M.W. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010.

[Lin and Kernighan, 1973] S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.

[Lusseau *et al.*, 2003] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, and S.M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[Newman, 2001] M. Newman. The structure of scientific collaboration networks. *Proceedings of National Academy of Sciences*, 98:404–409, 2001.

[Newman, 2006] M. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Ng *et al.*, 2002] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[Shi and Malik, 2000] J.B. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[Watts and Strogatz, 1998] D.J. Watts and S.H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):397–498, June 1998.

[Xu *et al.*, 2009] L. Xu, W. Li, and D. Schuurmans. Fast normalized cut with linear constraints. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2866–2873, 2009.

[Zachary, 1977] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.