Online Community Detection for Large Complex Networks

Wangsheng Zhang, Gang Pan*, **Zhaohui Wu, Shijian Li**
Department of Computer Science, Zhejiang University, China
{zws10, gpan, wzh, shijianli}@zju.edu.cn

Abstract

Complex networks describe a wide range of systems in nature and society. To understand the complex networks, it is crucial to investigate their internal structure. In this paper, we propose an online community detection method for large complex networks, which make it possible to process networks edge-by-edge in a serial fashion. We investigate the generative mechanism of complex networks and propose a split mechanism based on the degree of the nodes to create new community. Our method has linear time complexity. The method has been applied to six real-world network datasets and the experimental results show that it is comparable to existing methods in modularity with much less running time.

1 Introduction

Complex networks describe a wide range of systems in nature and society [Barrat *et al.*, 2004]. Frequently cited examples include the Internet, a network of routers and computers connected by physical links, and the citation network, a network of papers linked by citations. To understand the formation, evolution, and function of complex networks, it is crucial to investigate their internal structure, not only for uncovering the relations between internal structure and functions in complex networks, but also for practical applications in many disciplines such as biology and sociology [Ratti *et al.*, 2010; Szell *et al.*, 2012].

The problem of community detection is one of the outstanding issues in the study of network structure. A wide variety of community detection methods have been developed to serve different scientific needs. For example, [Ahn *et al.*, 2010] reinvent communities as groups of links rather than nodes and show that this unorthodox approach successfully reconciles the antagonistic organizing principles of overlapping communities and hierarchy. [Ziv *et al.*, 2005] propose a principled information-theoretic algorithm for community detection. However, when processing large complex networks, most existing community detection methods become impractical owing to the requirement that the whole network structure be available at each step of the method. For this reason, there has been a strong interest for online community detection method which make it possible to detect community without accessing the whole network structure at each step.

In this paper, we consider the generative mechanism of complex networks which has mainly not been considered in past studies and propose an online community detection method. Our method has $O(m)$ time complexity and $O(nk)$ space complexity. The results of applying our method and some other feasible methods to real-world network datasets suggest that our method is scalable and competitive in performance.

2 Related Work

Traditional methods of community detection, such as spectral bisection, the Kernighan-Lin algorithm and hierarchical clustering based on similarity measures are not ideal for the types of real-world network datasets with which current research is concerned [Newman, 2004b]. Modularity is a recently introduced quality measure for community detection. It was first proposed in [Newman, 2004b]. [Good *et al.*, 2010] describe the performance of modularity maximization in practical contexts and present a broad characterization of its performance in such situations. However, [Fortunato and Barthelemy, 2007] find that modularity optimization may fail to identify communities smaller than a scale which depends on the total size of the network and on the degree of interconnectedness of the communities. A check of the communities obtained through modularity optimization is thus necessary.

Modularity can be generalized in a principled fashion to incorporate the information contained in edge such as direction and weightiness. [Leicht and Newman, 2008] consider the problem of finding communities in directed networks. [Newman, 2004a] point out that weighted networks can in many cases be analyzed using a simple mapping from a weighted network to an unweighted multigraph. [Lancichinetti and Fortunato, 2009] generate directed and weighted networks with built-in community structure and show how modularity optimization performs on their benchmark.

A wide variety of modularity optimization methods have been developed [Leskovec *et al.*, 2010]. For example, [Clauset *et al.*, 2004] present a hierarchical agglomeration algorithm for detecting community. [Newman, 2006] show

*Author for correspondence

that the modularity can be expressed in terms of the eigenvectors of a characteristic matrix for the network and that this expression leads to a spectral algorithm for community detection. [Chen *et al.*, 2011] introduce a game-theoretic framework to address the community detection problem based on the social networks structure. [Li and Schuurmans, 2011] propose an iterative rounding strategy for identifying the communities that is coupled with a fast constrained power method that sequentially achieves tighter spectral relaxations. [Chan and Yeung, 2011] propose a convex relaxation scheme to give an iterative algorithm which solves the general k-partition problem. [Wu *et al.*, 2011] first projects node coordinates to the unit sphere and then applies the classic k-means to find communities. [Chen *et al.*, 2009] present a new community mining measure, max-min modularity, which considers both connected pairs and criteria defined by domain experts in finding communities, and then specify a hierarchical clustering algorithm to detect communities in networks. A recent comparative analysis of community detection algorithms is in [Lancichinetti and Fortunato, 2009].

3 Online Community Detection Method

3.1 Preliminaries

A network $G = \{V, E\}$ is a set of n nodes $V = \{v_1, v_2, \ldots, v_n\}$ connected by a set of m edges $E = \{e_{ij} = \{v_i, v_j\}\}$. The network considered here is undirected, unweighted and without self-loops or isolated node. Let $P = \{C_1, \ldots, C_K\}$ denote a partition of V, it is a division of V into K non-overlapping and non-empty communities C_k that cover all of V. As a performance measure for the quality of the partition, modularity was first proposed in [Newman, 2004b] and can be expressed as

$$q(P) = \sum_{C_k \in P} [\frac{edg(C_k)}{m} - (\frac{deg(C_k)}{2m})^2] \tag{1}$$

where $edg(C_k) = |\{e_{ij}|v_i \in C_k \text{ and } v_j \in C_k\}|$ is the number of intra-community edges within C_k and $deg(C_k)$ is the degree of C_k defined as $deg(C_k) = \sum_{v_i \in C_k} \deg(v_i)$ where $deg(v_i)$ is the degree of node v_i. Hence community detection can be formulated as an optimization problem

$$\max_P q(P) \tag{2}$$

and [Brandes *et al.*, 2008] prove the conjectured hardness of this problem both in the general case and with the restriction to number of partitions K.

3.2 Analysis of network expansion

Unlike most previous approaches, we consider networks as a result of expanding. It is growing by the addition of new edges. Given an existing network $G_{t-1} = \{V_{t-1}, E_{t-1}\}$, there are three cases for a new edge $e_t = \{v_i, v_j\}$ to be added in G_{t-1}, namely, (a)link a new node to an existing node, $\{v_i, v_j\} \cap V_{t-1} = \{v_i\}$ or $\{v_i, v_j\} \cap V_{t-1} = \{v_j\}$; (b)link two existing nodes, $\{v_i, v_j\} \subseteq V_{t-1}$; (c)link two new nodes, $\{v_i, v_j\} \cap V_{t-1} = \emptyset$(See Fig. 1).

Correspondingly, we do modularity optimization in a incremental way. For initial network $G_0 = \emptyset$, it is clearly that

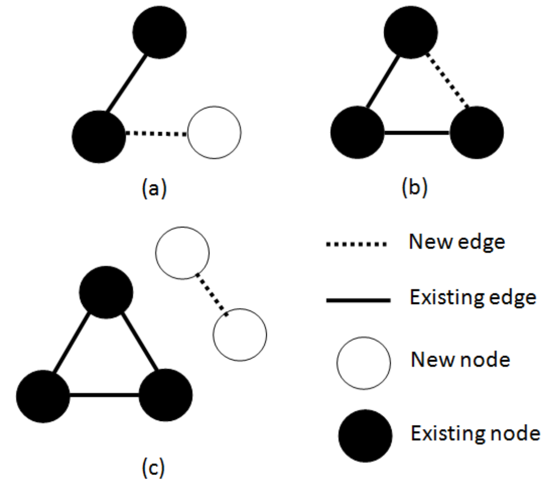

Figure 1: Three cases for a new edge to be added in an existing network: (a)link a new node to an existing node; (b)link two existing nodes; (c)link two new nodes

the best partition P is an empty set too. For subsequent networks $G_t, t = 1, 2, \ldots$, we derive our algorithm under those three cases separately as follow.

Case (a): link a new node to an existing node. Without loss of generality, we assume v_i is the existing node and v_j is the new node. It is easy to prove that a partition with maximum modularity has no community that consists of a single node with degree one. It seems that we should assign v_j to an existing community. However, this greedy approach will almost certainly lead to a local maximum. In worst case, all nodes are in a same community and result in zero modularity. Hence a split mechanism is necessary.

Let us recall the generative mechanism of complex networks. The first widely accepted model for the observed stationary scale-free distributions of complex networks was proposed by [Barabási and Albert, 1999] and its importance is recognized by academia [Newman, 2003; Boccaletti *et al.*, 2006]. It is based on two generic mechanisms: (a) networks expand continuously by the addition of new nodes; (b) new nodes attach preferentially to communities that are already well connected. Specifically, a new node v_j will attach an existing node v_i with probability $p(v_i)$ in proportion to the degree of node v_i.

$$p(v_i) \propto deg(v_i) \tag{3}$$

In other words, it is abnormal to observe a new node attaching an existing node with few edges and this phenomenon may indicate the emergence of a new community. This inspire us to split the new node to a new community with probability p_{split} that decreases with increases in the degree of the existing node. And the limit of p_{split}, as the degree of the existing node approaches infinite, should be zero. We choose p_{split} inversely proportional to the degree of the existing node.

$$p_{split} = \frac{1}{deg(v_i) + 1} \tag{4}$$

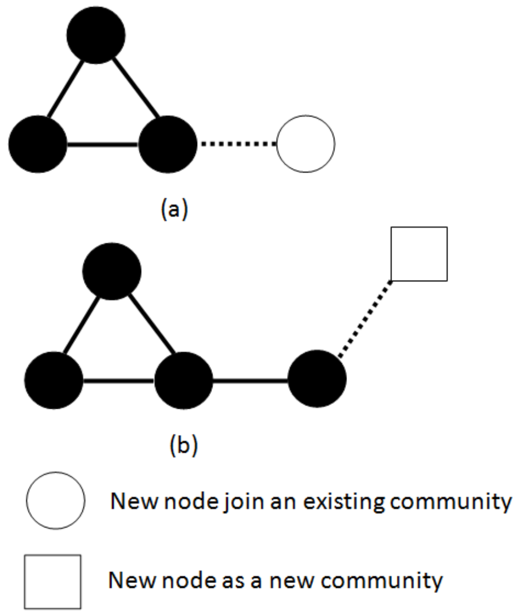

Figure 2: Split mechanism: (a)A new node attaches to an existing node with degree two, it join the same community of the existing node; (b)Another new node attaches to the previous new node with degree one, it split to a new community.

Here we use $deg(v_i)+1$ instead of $deg(v_i)$ to avoid deterministic split when $deg(v_i) = 1$. Fig. 2 depict our split mechanism. When a new node attaches to an existing node with degree two, it join the same community of the existing node; when another new node attaches to the previous new node with degree one, it split to a new community.

For a complete review of the statistical mechanics of network topology and dynamics of complex networks, one can refer to [Albert and Barabási, 2002]. And [Mitzenmacher, 2004] briefly survey some other generative models that lead to scale-free distributions.

Case (b): link two existing nodes. Existing nodes may belong to a same community or not(See Fig. 3). If both nodes belong to a same community, the new edge just increase the number of intra-community edges in the community so we do nothing. If they belong to different communities, we evaluate $\Delta q(v_i, v_j)$, which is defined as modularity gain for node v_i moving from it's community $C_{k(i)}$ to node v_j's community $C_{k(j)}$, and $\Delta q(v_i, v_j)$'s counterpart $\Delta q(v_j, v_i)$. If either of them larger than zero, we move one node according to maximum modularity gain principle.

Notice that $\Delta q(v_i, v_j)$ can be calculated in constant time through follow equation

$$\begin{aligned}\Delta q(v_i, v_j) &= q(P') - q(P)\\
&= \sum_{C'_k \in P'} [\frac{edg(C'_k)}{m} - (\frac{deg(C'_k)}{2m})^2]\\
&- \sum_{C_k \in P} [\frac{edg(C_k)}{m} - (\frac{deg(C_k)}{2m})^2]\\
&= [\frac{edg(C'_{k(i)})}{m} - (\frac{deg(C'_{k(i)})}{2m})^2]\\
&+ [\frac{edg(C'_{k(j)})}{m} - (\frac{deg(C'_{k(j)})}{2m})^2]\\
&- [\frac{edg(C_{k(i)})}{m} - (\frac{deg(C_{k(i)})}{2m})^2]\\
&- [\frac{edg(C_{k(j)})}{m} - (\frac{deg(C_{k(j)})}{2m})^2]\\
&= \frac{deg(v_i, C_{k(j)}) - deg(v_i, C_{k(i)})}{m}\\
&+ \frac{[edg(C_{k(i)}) - edg(C_{k(j)}) - deg(v_i)]deg(v_i)}{2m^2}
\end{aligned} \quad (5)$$

where $deg(v_i, C_k) = |\{e_{ij}|v_j \in C_k\}|$ is number of edges from the node v_i to community C_k and $C_{k(i)}$ is the community which the node v_i belongs to.

We can improve modularity via other types of adjustments. For example, the modularity gain of merging two communities $C_{k(i)}$ and $C_{k(j)}$ is

$$\begin{aligned}\Delta q(C_{k(i)}, C_{k(j)}) = &\frac{deg(C_{k(i)}, C_{k(j)})}{m}\\
&- \frac{deg(C_{k(i)})deg(C_{k(j)})}{2m^2}\end{aligned} \quad (6)$$

where $deg(C_{k(i)}, C_{k(j)})$ is number of edges between community $C_{k(i)}$ and community $C_{k(j)}$. Sometimes $\Delta q(C_{k(i)}, C_{k(j)})$ can be positive and significantly larger than $\Delta q(v_j, v_i)$ and $\Delta q(v_i, v_j)$. However, merging two communities may cause that subsequent optimazitions suffer from overlarge community size. For the sake of simplicity, we do not take those complex adjustments.

Case (c): link two new nodes. As a partition of maximum modularity does not include disconnected communities, those two nodes should not join any existing communities but form a new community themselves.

3.3 The proposed algorithm

Taken together, our method take a sequence of edges as input and do modularity optimization incrementally. If only one node of current edge belongs to the existing network, we split another node to a new community with probability inversely proportional to the degree of the existing node; If both nodes of current edge belong to the existing network but they belong to different communities, we move one node according to maximum modularity gain principle; If neither node of current edge belongs to the existing network, we just assign them

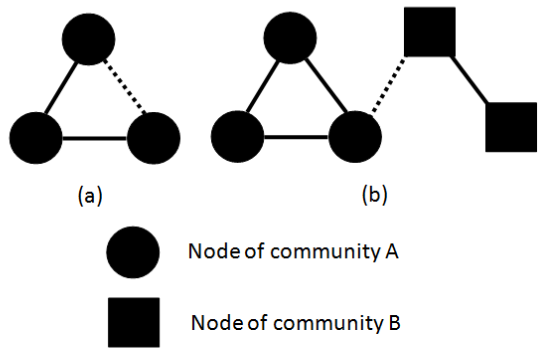

Figure 3: Two situations of a new edge link two existing nodes. (a)Nodes belong to a same community; (b)Nodes belong to different communities.

to a new community. Obviously, our online community detection algorithm has $O(m)$ time complexity. And the space complexity is $O(nk)$ because we need to store $deg(v_i, C_k)$ for calculating modularity gain in constant time. The algorithm is summarized in Algorithm 1.

4 Experiments

In this section, we present experimental results of our online community detection method and compare it with some other methods. For simplicity, we use *OL* to refer to our method, *CNM* to refer to an agglomerative method [Clauset *et al.*, 2004] and *Eig* to refer to the eigenvector-based method [Newman, 2006].

We use six real-world network datasets from Stanford Large Network Dataset Collection[1], which commonly used by other researchers for community detection, their edge sizes varying from 93,439 to 1,992,636(See Table 1). These datasets are

- **ca-CondMat**: Collaboration network of Arxiv Condensed Matter [Leskovec *et al.*, 2007];
- **ca-HepPh**: Collaboration network of Arxiv High Energy Physics [Leskovec *et al.*, 2007];
- **ca-AstroPh**: Collaboration network of Arxiv Astro Physics [Leskovec *et al.*, 2007];
- **cit-HepTh**: Arxiv High Energy Physics paper citation network [Leskovec *et al.*, 2005];
- **cit-HepPh**: Arxiv High Energy Physics paper citation network [Leskovec *et al.*, 2007]
- **web-Stanford**: Web graph of Stanford.edu [Leskovec *et al.*, 2009].

The edges should be processed in order of creation. However, the datasets do not have timestamps on the edges. We process the edges in order of their appearance in the raw dataset file.

[1]http://snap.stanford.edu/data/

Algorithm 1: Online Community Detection Algorithm

```
Input: a sequence of edges {e_1, ..., e_t, ..., e_T}
Output: partition P = {C_1, ..., C_K}
V_0 ← ∅, E_0 ← ∅, P ← ∅, K ← 0;
for t ← 1 to T do
    fetch e_t = {v_i, v_j};
    V_t ← V_{t-1} ∪ {v_i, v_j}, E_t ← E_{t-1} ∪ {e_t};
    if {v_i, v_j} ∩ V_{t-1} = ∅ then
        C_{K+1} ← {v_i, v_j};
        P ← P ∪ {C_{K+1}};
        K ← K + 1;
    else
        if {v_i, v_j} ⊆ V_{t-1} then
            if C_{k(i)} ≠ C_{k(j)} then
                if max(Δq(v_i, v_j), Δq(v_j, v_i)) > 0 then
                    if Δq(v_i, v_j) > Δq(v_j, v_i) then
                        C_{k(i)} ← C_{k(i)}/{v_i};
                        C_{k(j)} ← C_{k(j)} ∪ {v_i};
                    else
                        C_{k(j)} ← C_{k(j)}/{v_j};
                        C_{k(i)} ← C_{k(i)} ∪ {v_j};
        else
            if {v_i, v_j} ∩ V_{t-1} = {v_i} then
                draw x from Bernoulli(1/(deg(v_i)+1));
                if x = 0 then
                    C_{k(i)} ← C_{k(i)} ∪ {v_j};
                else
                    C_{K+1} ← {v_j};
                    P ← P ∪ {C_{K+1}};
                    K ← K + 1;
            else
                draw x from Bernoulli(1/(deg(v_j)+1));
                if x = 0 then
                    C_{k(j)} ← C_{k(j)} ∪ {v_i};
                else
                    C_{K+1} ← {v_i};
                    P ← P ∪ {C_{K+1}};
                    K ← K + 1;
return P;
```

Network	Node Size	Edge Size
ca-CondMat	23,133	93,439
ca-HepPh	12,008	118,489
ca-AstroPh	18,772	198,050
cit-HepTh	27,770	352,285
cit-HepPh	34,546	420,877
web-Stanford	281,903	1,992,636

Table 1: Summary of network datasets

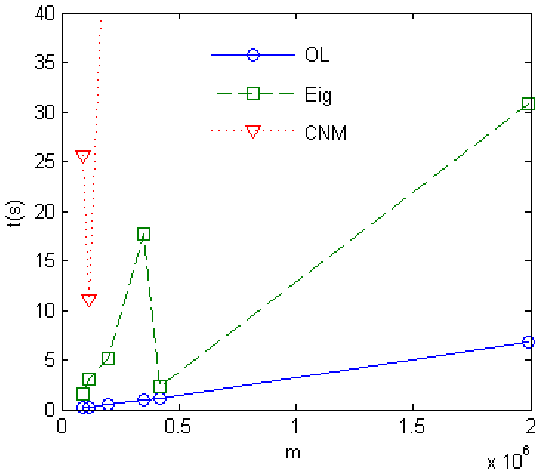

Figure 4: Running time t(in seconds) of six datasets with different edge size m by *OL*, *Eig* and *CNM*.

We use C# to implement our algorithm[2]. For comparison, we employ implementation of *CNM* algorithm and *Eig* algorithm in igraph package[3] in R software[4]. Both *CNM* algorithm and *Eig* algorithm are written in C language. We carry out experiments on a Windows based Genuine Intel(R) CPU i3 @ 3.20GHz machine with 4.00GB memory.

Maximum and average modularity and variance of modularity over 10 runs by *OL* as well as modularity over 10 runs by *Eig* and *CNM* are reported in Table 2. We can see that *OL* outperforms Eig consistently for all six datasets and outperforms *CNM* except the last dataset, and the gap between *OL* and *CNM* in the last dataset is less than 0.02. The variance is low, so we need not run multiple times.

Average running time(in seconds) over 10 runs by *OL*, *Eig* and *CNM* are reported in Table 3 and Fig. 4. *OL* also has significantly advantage in terms of running time. It is linear in edge size as we expected. For networks with two millions of edges, it can give result in six seconds which is six hundred times faster than *CNM*.

To evaluate the convergence speed of *OL*, we plot the average temporal modularity over 10 runs by *OL*(See Fig. 5). We can see that *OL* can give a acceptable modularity immediately after process start and the modularity becomes stable in early stage for all six datasets.

5 Conclusion

In this paper we have examined the problem of detecting community in large complex networks, which is formulated as an optimization problem in which one searches for the maximum of the quantity known as modularity over possible partition of a network. We have considered the generative mechanism of complex networks and presented a new method which allows us to perform online modularity maximization. The method has been applied to a variety of real-world network datasets and our experiments give very encouraging results. Not only is the proposed method scalable in terms of both time and space complexity, but it also gives competitive performances. Our future research will consider the use of quality measures other than modularity for solving the community detection problem under an online optimization framework, as well as apply the method to directed and weighted complex networks.

[2]Our C# implementation can be downloaded from http://www.cs.zju.edu.cn/people/gpan/

[3]http://igraph.sourceforge.net/

[4]http://www.r-project.org/

Acknowledgments

The authors want to thank anonymous reviewers for helpful comments. This work was supported by National Key Basic Research Program of China(2013CB329504) and Qianjiang Talent Program of Zhejiang(2011R10078).

References

[Ahn *et al.*, 2010] Y.Y. Ahn, J.P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[Albert and Barabási, 2002] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

[Barabási and Albert, 1999] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[Barrat *et al.*, 2004] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752, 2004.

[Boccaletti *et al.*, 2006] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: structure and dynamics. *Physics Reports*, 424(4):175–308, 2006.

[Brandes *et al.*, 2008] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[Chan and Yeung, 2011] E.Y.K. Chan and D.Y. Yeung. A convex formulation of modularity maximization for community detection. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2218–2225. AAAI Press, 2011.

[Chen *et al.*, 2009] Jiyang Chen, Osmar R. Zaïane, and Randy Goebel. Detecting communities in social networks using max-min modularity. In *SDM*, pages 978–989, 2009.

[Chen *et al.*, 2011] W. Chen, Z. Liu, X. Sun, and Y. Wang. Community detection in social networks through community formation games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2576–2581. AAAI Press, 2011.

[Clauset *et al.*, 2004] A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

Network	MAX(q)[OL]	AVG(q)[OL]	VAR(q)[OL]	q[Eig]	q[CNM]
ca-CondMat	0.657	0.654	3.237×10^{-6}	0.251	0.626
ca-HepPh	0.608	0.605	5.726×10^{-6}	0.571	0.578
ca-AstroPh	0.566	0.558	3.518×10^{-5}	0.441	0.503
cit-HepTh	0.620	0.616	2.925×10^{-5}	0.494	0.505
cit-HepPh	0.682	0.671	4.180×10^{-5}	0.000	0.557
web-Stanford	0.875	0.872	6.717×10^{-6}	0.050	0.894

Table 2: Maximum and average modularity and variance of modularity over 10 runs by *OL* as well as modularity over 10 runs by *Eig* and *CNM*.

Network	AVG(t)[OL]	AVG(t)[Eig]	AVG(t)[CNM]
ca-CondMat	0.278	1.5	25.6
ca-HepPh	0.232	3.0	11.1
ca-AstroPh	0.535	5.1	57.8
cit-HepTh	0.897	17.7	138.1
cit-HepPh	1.186	2.3	218.2
web-Stanford	6.802	30.8	3605.6

Table 3: Average running time(in seconds) over 10 runs by *OL*, *Eig* and *CNM*.

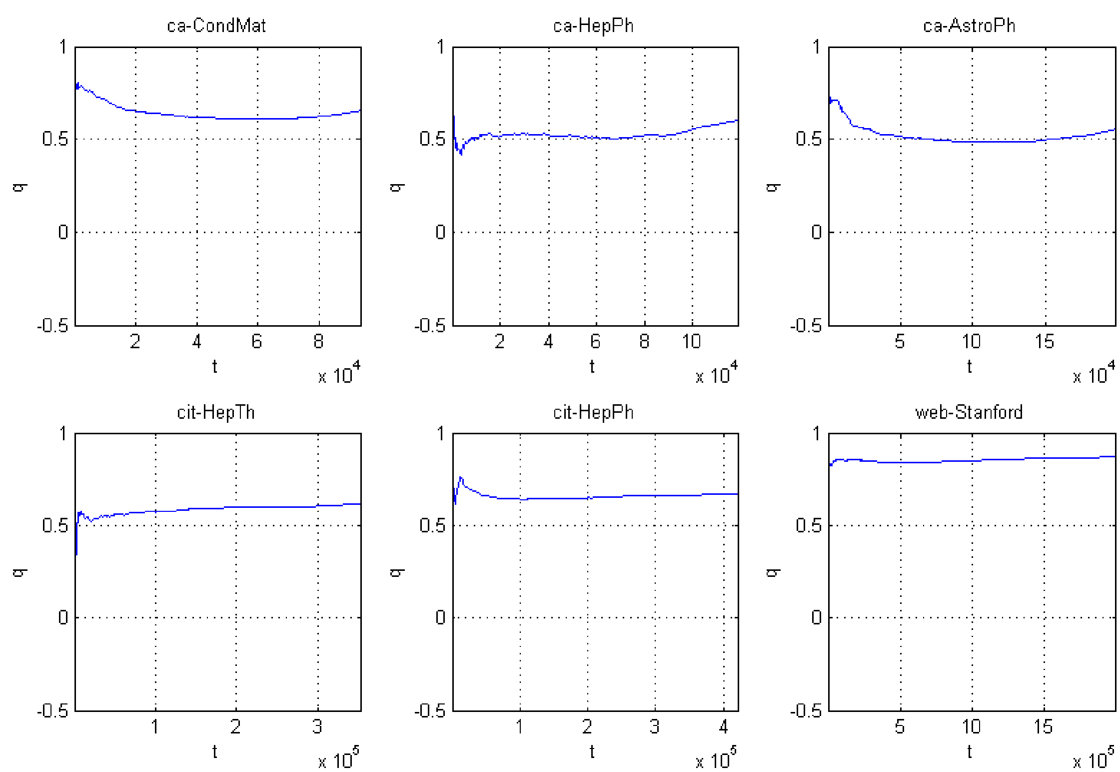

Figure 5: Average temporal modularity over 10 runs by *OL*.

[Fortunato and Barthelemy, 2007] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[Good *et al.*, 2010] B.H. Good, Y.A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[Lancichinetti and Fortunato, 2009] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[Leicht and Newman, 2008] E.A. Leicht and M.E.J. Newman. Community structure in directed networks. *Physical Review Letters*, 100(11):118703, 2008.

[Leskovec *et al.*, 2005] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187. ACM, 2005.

[Leskovec *et al.*, 2007] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[Leskovec *et al.*, 2009] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[Leskovec *et al.*, 2010] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the Nineteenth International Conference on World Wide Web*, pages 631–640. ACM, 2010.

[Li and Schuurmans, 2011] W. Li and D. Schuurmans. Modular community detection in networks. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1366–1371. AAAI Press, 2011.

[Mitzenmacher, 2004] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251, 2004.

[Newman, 2003] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[Newman, 2004a] M.E.J. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.

[Newman, 2004b] M.E.J. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

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

[Ratti *et al.*, 2010] C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton, and S.H. Strogatz. Redrawing the map of great britain from a network of human interactions. *PLoS One*, 5(12):e14248, 2010.

[Szell *et al.*, 2012] M. Szell, R. Sinatra, G. Petri, S. Thurner, and V. Latora. Understanding mobility in a social petri dish. *Scientific Reports*, 2, 2012.

[Wu *et al.*, 2011] L. Wu, X. Ying, X. Wu, and Z.H. Zhou. Line orthogonality in adjacency eigenspace with application to community partition. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2349–2354. AAAI Press, 2011.

[Ziv *et al.*, 2005] E. Ziv, M. Middendorf, and C.H. Wiggins. Information-theoretic approach to network modularity. *Physical Review E*, 71(4):046117, 2005.