Available online at www.sciencedirect.com
ScienceDirect
Physica A 377 (2007) 363–372
PHYSICA A
www.elsevier.com/locate/physa

Accuracy and precision of methods for community identification in weighted networks

Ying Fan[a,*], Menghui Li[a], Peng Zhang[a], Jinshan Wu[b], Zengru Di[a]

[a]*Department of Systems Science, School of Management, Beijing Normal University, Beijing 100875, PR China*
[b]*Department of Physics and Astronomy, University of British Columbia, Vancouver, B.C. Canada, V6T 1Z1*

Received 2 October 2006; received in revised form 31 October 2006
Available online 8 December 2006

Abstract

Different algorithms, which take both links and link weights into account for the community structure of weighted networks, have been reported recently. Based on the measure of similarity among community structures introduced in our previous work, in this paper, accuracy and precision of three algorithms are investigated. Results show that Potts model based algorithm and weighted extremal optimization (WEO) algorithm work well on both dense or sparse weighted networks, while weighted Girvan–Newman (WGN) algorithm works well only for relatively sparse networks.

Keywords: Weighted networks; Community structure; Similarity function

1. Introduction

In recent years, more and more systems in many different fields are depicted as complex networks. Recent empirical studies on networks display that there are communities in social networks, metabolic networks, economic networks [1–6] and so on. The community structure is an important character to understand the functional properties of complex networks. For instance, in food web, communities reveal the subsystems of ecosystem [7]. In the world wide web, the community analysis has found thematic groups [8,9]. E-mail network can be divided into departmental groups whose work is distinct and the communities reveal organization structures [10,11]. In biochemical or neural networks, communities may correspond to functional groups [5]. The deep understanding on community structure will make us comprehend and analyze the characteristics of systems better.

Intuitively, community structure in binary networks is groups of network vertices where links within same group are much denser than links between different groups. In order to evaluate community structure in networks, the modularity, as a mathematic form, is suggested by Girvan and Newman [12], then a fast algorithm based on maximizing the modularity Q is proposed [13]. Binary networks focus on the existence of connections and neglect the strength of links. However, it has been widely recognized that many real networks

*Corresponding author.
E-mail address: yfan@bnu.edu.cn (Y. Fan).

0378-4371/$ - see front matter
doi:10.1016/j.physa.2006.11.036

are intrinsically weighted. Link weights (strength of connections) indicate particularly close connections or similarity between vertices. Obviously, strength of connections has important effect on community structure. Given the same topological structure, different assignments of link weights may result in different community structures. For example, consider the family structure of human society, say a, b form a family ab and a/b's parents and brothers/sisters form family A/B. Ignoring the strength of connections we will naturally get two groups $a + A$ and $b + B$. We are not saying the above structure is wrong, but not as detailed and clear as the three group structure A, B, ab, which can only be shown after considering strength of connections. Therefore pursuing algorithms detecting community structure for weighted networks is a valuable task. And community structure of weighted networks could give us better understanding for real functional groups. In a sense, detecting weighted community structure is to maximize the weighted modularity Q^w [13]:

$$Q^w = \frac{1}{2T}\sum_{ij}\left[w_{ij} - \frac{T_i T_j}{2T}\right]\delta(c_i, c_j), \tag{1}$$

where w_{ij} represents the link weight (similarity weight) between nodes i and j, T_i is the vertex weight of node i: $T_i = \sum_j w_{ij}$, $T = \frac{1}{2}\sum_{ij} w_{ij}$ is the sum of all link weights in networks, and c_i shows that vertex i belongs to community c_i.

Many algorithms for community identification have been proposed recently [1,13–28]. They detect communities according to topological structures or dynamical behaviors of networks. Most of them are developed for binary networks. But some of them can be generalized to weighted networks. For example, in its binary version, GN algorithm [1] is based on the link betweenness. However, when the link weight is taken into account, Newman has already generalized GN algorithm [29], which adds only the extra operation of dividing the unweighted link betweenness by the link weight. Similarly, binary Potts model based algorithm [24] can be generalized to its weight version by relating coupling strength J_{ij} to similarity link weight w_{ij}. And extremal optimization (EO) [25] can be generalized to weighted extremal optimization (WEO) algorithm, when the object function modularity Q is replaced by the weighted modularity Q^w.

However, since there are so many methods proposed, we should ask that which method is more appropriate for detecting communities. There have been discussions on sensitivity and computational cost of those algorithms on binary networks (see Ref. [30] and its references). Here we are going to discuss performance of their weighted version, namely weighted GN (WGN) algorithm [29], weighted Potts model and (WEO) methods. These three algorithms stand for three different approaches: topological structure based (WGN), dynamics based, and modularity based. We will discuss these three algorithms briefly in the next section.

We define accuracy and precision of a method to evaluate the performance of an algorithm. As what have been done by previous studies, we can apply these approaches on some ad hoc networks with a well known, fixed community structures [1]. Then accuracy means the consistence when the community structure from algorithm is compared with the presumed communities, and precision is the consistence among the community structures from different runs of an algorithm on the same network. The question about how to quantitatively measure the difference between two community structures has already be discussed before. Newman described a method to calculate the sensitivity of algorithms [13]. Danon et al proposed a measurement $I(A, B)$ based on information theory [30]. These two measurements mainly focus on the proportion of nodes which are correctly grouped. We proposed a similarity function S to measure the difference between partitions [31]. The function S integrates the information about the proportion of nodes co-appearance in pair groups of A, B and the total number of communities. Starting from two community structures $\{A_1, A_2, \ldots A_K\}$ and $\{B_1, B_2, \ldots B_M\}$ over the same set N, firstly, we can get similarities of every pair of groups by similarity function,

$$s_{ij} = \frac{|A_i \cap B_j|}{|A_i \cup B_j|}. \tag{2}$$

For each group in community A, we can find its counterpart in community B by the maximum of similarity measurement. It identifies the correspondence between As and Bs. Then, for each pair of groups, the similarity s_l of A_l and B_l can be given by Eq. (2). The number of communities is usually different. It means some A_l do not have any counterparts. In this case, the counterparts of these communities are considered as an empty set Φ, so s_l is equal to 0. And the total similarity can be calculated as

$$S = \frac{\sum_{l=1}^{\max(K,M)} s_l}{\max(K, M)}. \tag{3}$$

Here, we use the similarity function S to quantify the difference of different partitions.

This paper is organized as following. To draw a general conclusion and know the conditions effecting the performance of algorithms, we need some weighted networks with well-known community structures. In the next section, we generate lots of idealized networks to discuss the problem systematically. After introducing the algorithms briefly in Section 2.1, we present the performance of the binary version of those three algorithms in Section 2.2. The results consistent with the previous studies. That suggests the similarity function serves well as the measurement of difference between communities. Then we generate more ideal ad hoc weighted networks with different strength of connections to compare those three algorithms more throughout. The conclusion confirms that WGN performs well when binary connections is clear enough to show the community structure, while Potts and WEO are more robust even when the community structure is not clear when only binary connections are considered. In Section 2.4, we use an extreme case, a complete graph but with different link weights, to discuss the accuracy and precision of those algorithms. From the above investigations, we find that except WGN algorithm, a high precision of Potts and WEO algorithm implies the existence of community structure. So the concept of precision for Potts and WEO algorithm is not only just for the discussion of algorithm performance, but also an indicator of the existence of community structure. Then in Section 3, we apply above algorithms on Rhesus monkey network to get some empirical results. The community structure of the monkey network has already been studied by Newman [29] through WGN algorithm. Here we run Potts and WEO algorithms on it and compare their accuracies and precisions. The results reveal again that for such a dense weighted network, Potts and WEO perform better than WGN. Some remarks are given in Section 4.

2. Results based on idealized ad hoc networks

2.1. Brief introduction for the methods of community identification

In this paper, we mainly discuss the performance of WGN algorithm, Potts model algorithm, and WEO algorithm in weighted networks.

The WGN algorithm is based on the concept of edge betweenness. The betweenness of an edge in network, generalized based on Freemans betweenness centrality [32], is defined as the number of the shortest paths passing through it [1]. It is very clear that edges, which connect communities, have a larger betweenness value, as all shortest paths that connect nodes in different communities have to run along it. By removing the edge with the largest betweenness at each step, the whole network can be gradually split into isolated components or communities [1]. Taking link weights into account, this idea has been generalized to weighted networks [29] as following. Firstly, we calculate the link betweenness B_{ij} of all edges in nonweight networks. Then we divide each betweenness B_{ij} by link weight w_{ij} of the corresponding edge. Next, we remove the link with the highest value $\frac{B_{ij}}{w_{ij}}$, recalculate the betweenness, and repeat till all edges are removed. In addition, we also use weighted modularity Q^w (Eq. (1)) to judge community divisions in weighted networks.

For Potts model algorithm, the community coincides with the domain of equal spin value in the minima of modified Potts spin glass Hamiltonian. The node in networks just be looked as the electron with spin elected from q spin states. The link correspond to the reciprocity between electrons. This will allow us to partition the communities of a network onto the magnetic domains in the ground state or in local minima of a suitable Hamiltonian. For this purpose, a global constraint that forces the spins into communities is appended to the q-state Potts Hamiltonian [24],

$$H = \sum_{(i,j)\in E} J_{ij}\delta_{\sigma_i\sigma_j} + \gamma \sum_{s=1}^{q} \frac{n_s(n_s - 1)}{2}, \tag{4}$$

where, σ_i denotes the individual spins which are allowed to take q values $1 \dots q$, n_s denotes the number of spins that have spin s such that $\sum_{s=1}^{q} n_s = N$, J_{ij} is the ferromagnetic interaction strength, $\gamma = \frac{2\langle J_{ij}\rangle M}{N(N-1)}$ is a positive parameter determined by total number of nodes N and total number of links M. To practically find or approximate the ground state of system, a simple Metropolis algorithm could be employed. The system is cooled down using a decrement function for the temperature of the form $T_{k+1} = \alpha T_k$ with $\alpha = 0.99$. For weighted networks, J_{ij} can be taken as similarity link weight. Then Potts model algorithm can be easily applied to weighted networks.

The EO algorithm uses a heuristic search to optimize the modularity Q by dividing network into groups [25]. When this method is generalized to weighted networks, Q is replaced by Q^w as the global variable to optimize. Rewriting Eq. (1) as

$$Q^w = \sum_r (e_{rr}^w - (a_r^w)^2), \tag{5}$$

where $e_{rr}^w = \frac{1}{2T}\sum_{ij} w_{ij}\delta(c_i, r)\delta(c_j, r)$ is the fraction of summation of link weight that connect two nodes inside the community r, $a_r^w = \frac{1}{2T}\sum_i T_i \delta(c_i, r)$ is the fraction of summation of vertex weight of community r. While the value of each node contributing to weighted modularity Q^w can be defined as

$$q_i^w = T_{r(i)} - T_i a_{r(i)}^w, \tag{6}$$

where $T_{r(i)}$ is the summation of link weight that a node i belonging to a community r has with nodes in the same community, and T_i is the vertex weight of node i. So the modularity Q^w is $Q^w = \frac{1}{2T}\sum_i q_i^w$. Rescaling the local variable q_i^w by the vertex weight of node i, the contribution of node i to the weighted modularity is defined as

$$\lambda_i^w = \frac{q_i^w}{T_i} = \frac{T_{r(i)}}{T_i} - a_{r(i)}^w \tag{7}$$

λ_i^w is normalized in the interval $[-1, 1]$. It gives the relative contributions of individual nodes to the community structure. So it could be considered as the fitness of a node involved in the WEO process. The process of detecting community structure by WEO is as follows:

1. Initially, split randomly the whole network into two groups with similar number of nodes.
2. At each time step, move the node with the lower fitness from one group to the other. After each movement, recalculate the fitness of every node based on Eq. (7).
3. Repeat process 2 until a maximum value of Q^w is reached. After that, proceed recursively with every group. When the modularity Q^w cannot be improved, the process will finish.

In order to escape from local maxima, WEO algorithm adopts τ-EO method [33]. The node is selected according to the following probability:

$$P(q) \propto q^{-\tau}, \tag{8}$$

where q is the rank number of node according to their fitness values, and $\tau{\sim}1 + 1/\ln(N)$ is related with the network size N.

Recently, Reichardt and Bornholdt indicates that Potts Hamiltonian is a special cases of the modularity (Eq. (1)) at some conditions [34],

$$Q^w = -\frac{1}{M} H\{\sigma\}. \tag{9}$$

So even the motivation of above two algorithms are different, the Potts model algorithm and the EO algorithm are both algorithms that maximize modularity. The difference between them is only that they use different algorithms to find the maximum modularity. Potts model algorithm is a thermal exploration (Metropolis or Monte-Carlo heat-bath algorithm with simulated annealing). While the EO algorithm explores the maxima of global variable by moving the node with the lower fitness (extremal) from one partition to the other directly.

2.2. Results on binary ad hoc networks

Discussion in this subsection is totally not new. It is rather a repeat of works done in Ref. [30], but here we use similarity measurement S to discuss the accuracy and precision. However, by doing so we notice two things that first our measure S services well for its purpose and community structure becomes ambiguous when k_{inter} is too large.

In order to investigate the performance of algorithms, ad hoc networks is firstly introduced by Newman [1] and used by many other authors [24,25,30]. Each network consists of $n = 128$ vertices, which divided into four groups of 32 nodes. Vertices are assigned to groups and are randomly connected to vertices of the same group by an average of $\langle k_{intra} \rangle$ links and to vertices of different groups by an average of $\langle k_{inter} \rangle$ links. The average degree of all vertices are fixed, namely $\langle k_{intra} \rangle + \langle k_{inter} \rangle = 16$. It is obvious that with $\langle k_{inter} \rangle$ increasing, the communities become more diffuse and it becomes more difficult to detect the communities.

From Fig. 1 we can see that for small $\langle k_{inter} \rangle$, there are no discrimination among results from WGN, Potts and WEO algorithms. But for large $\langle k_{inter} \rangle$, WGN is stable but its accuracy drops quite quickly. However, Potts model and WEO algorithm are fluctuant when $\langle k_{inter} \rangle$ is large, though their accuracy are acceptable even for quite large $\langle k_{inter} \rangle$.

2.3. Results on weighted ad hoc networks

In weighted networks, we use similarity link weight to describe the closeness of relations between vertices. The larger the link weight is, the closer the relation is. Under the basic construction of ad hoc network described above, the intragroup link weight is assigned as w_{intra}, while the intergroup link weight is assigned as w_{inter}. In practice, the relationship among the nodes in groups is usually much closer than the relationship between groups. So w_{inter} is normally less than w_{intra}. Similarly with $\langle k_{intra} \rangle + \langle k_{inter} \rangle = 16$, we require the link weight on intra and inter links follow the constraint:

$$\langle w_{intra} \rangle + \langle w_{inter} \rangle = 2, \tag{10}$$

where $\langle w_{intra} \rangle$ ($\langle w_{inter} \rangle$) is the average of all intragroup (intergroup) link weights. Usually, link weights obey a distribution, even a special kind of distribution, for example, a power-law distribution. Here for simplicity, we assign the same weight $w_{inter} = w$ to all intergroup links, and assign the same weight $w_{intra} = 2 - w$ to all

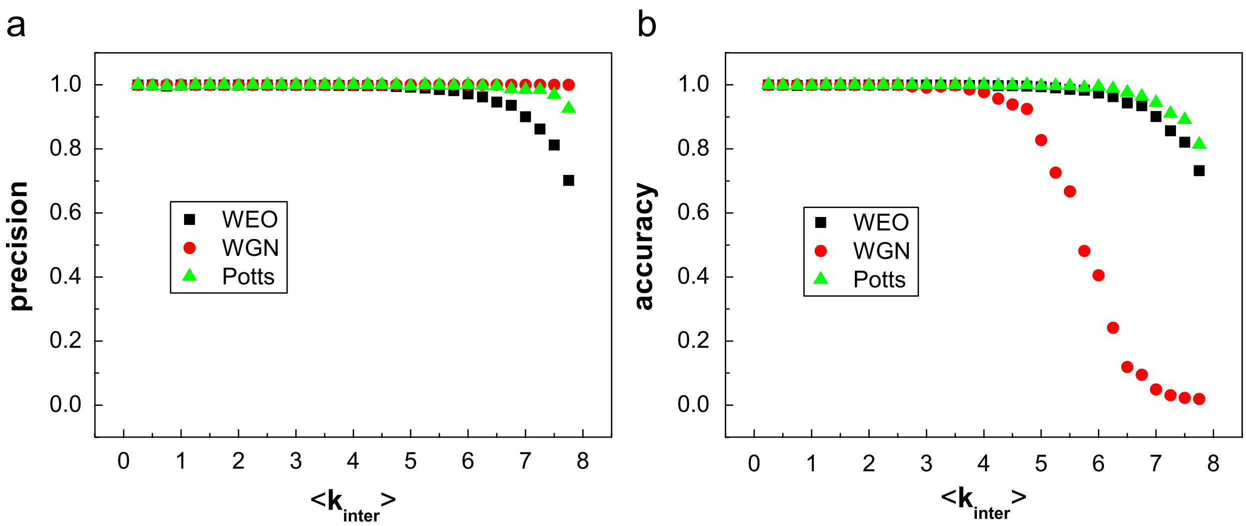

Fig. 1. Algorithm performance as applied to ad hoc networks with $n = 128$ and four communities of 32 nodes each. Total average degree is fixed to 16. (a) Comparing precision of the algorithm by several results on same ad hoc networks. (b) Comparing accuracy using ad hoc networks with presumed community structure. The x-axis is the average of connections between nodes in different groups $\langle k_{inter} \rangle$. Each point is an average over 20 networks and 10 runs each.

intragroup links. This means our above equation just simply becomes

$$w_{intra} + w_{inter} = 2. \tag{11}$$

Later we also try the uniform distribution cases. However, we will see that results from this case have no qualitative difference with the one with δ distribution.

In the following numerical investigations, we first get 20 realizations of idealized ad hoc networks under the same conditions. Then we run each algorithm to find communities in each network 10 times. Based on these results, using the similarity function S, comparing each pair of these 10 community structures and averaging over the 20 networks (average of totally $C_{10}^2 \times 20 = 900$ results) we could get the precision of the algorithm. Comparing each divided groups with the presumed structures, we can get the accuracy of the algorithm by averaging these $10 \times 20 = 200$ results.

When $\langle k_{inter} \rangle$ is small, community structure can be detected just from its binary correspondence of the weighted networks—simply its binary network when mentioned in the future. In this case, we know that the community structure is dominated by links. Therefore, in this case, when $w_{inter} \leqslant 1$, there is no ambiguity in community structure. In other hand, when $\langle k_{inter} \rangle$ is too large, the communities is very diffuse in binary networks and of course it is impossible to find communities correctly by any algorithms. So the interesting things happen when $\langle k_{inter} \rangle \approx \langle k_{intra} \rangle$, where the link weights play a crucial role in the partition of communities. Hopefully, one would like to predict that when $w_{inter} < w_{intra}$ the community structure should still be clear. Therefore, if algorithm is good then precision and accuracy of the community structure should still be good. However, we find from Fig. 2 that accuracy of WGN decrease much faster than the others although its precision stays almost constant. We can see that even when w_{inter} around 0.8 both precision and accuracy of Potts model and WEO algorithms are still quite good.

Another interesting case is $\langle k_{inter} \rangle < \langle k_{intra} \rangle$ but $\langle w_{inter} \rangle > \langle w_{intra} \rangle$, where information from connection and weight are in some kinds of contradiction. This is investigated in Fig. 3 along with the trivial case that $\langle k_{inter} \rangle < \langle k_{intra} \rangle$ and $\langle w_{inter} \rangle < \langle w_{intra} \rangle$.

Next we use ad hoc networks with uniform distribution of link weights. For a given network topology with certain $\langle k_{inter} \rangle$, weights are taken randomly from the interval $[\langle w_{intra} \rangle - 0.25, \langle w_{intra} \rangle + 0.25]$ and $[\langle w_{inter} \rangle - 0.25, \langle w_{inter} \rangle + 0.25]$, respectively, for intragroup connections and intergroup connections. With changing its average value, we can also get the performance of algorithms under different conditions. The results are summarized in Fig. 4. They are qualitatively similar with the above results.

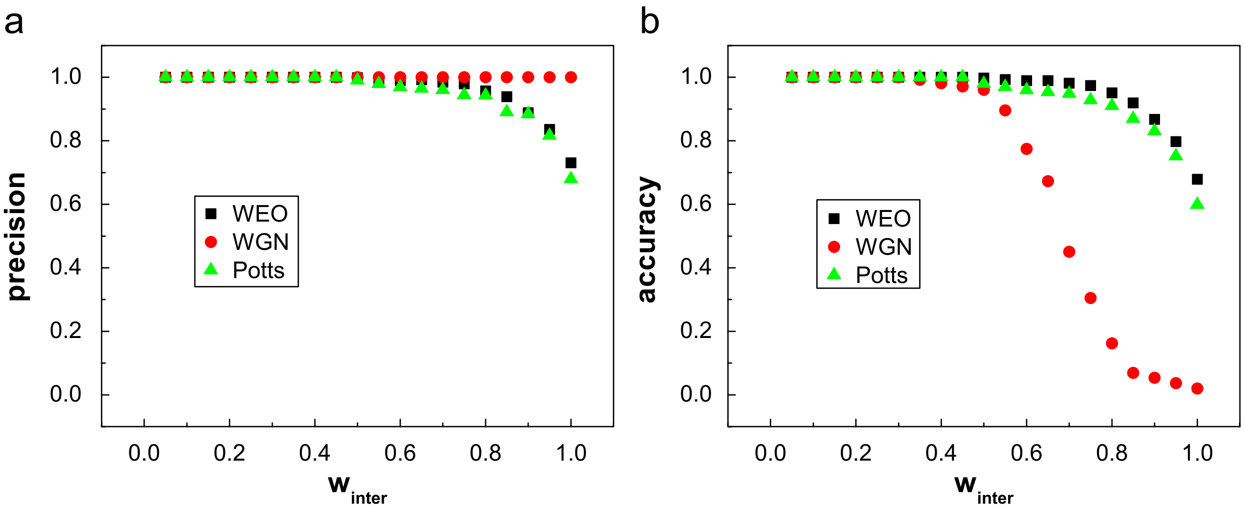

Fig. 2. The influence of weight on community structure when $\langle k_{inter} \rangle = \langle k_{intra} \rangle = 8$, where link weight could play crucial role. w_{inter} changes from 0.05 to 1. We see that accuracy of WGN drops much faster than the others, and precision and accuracy of Potts and WEO are still good around $w_{inter} = 0.8$.

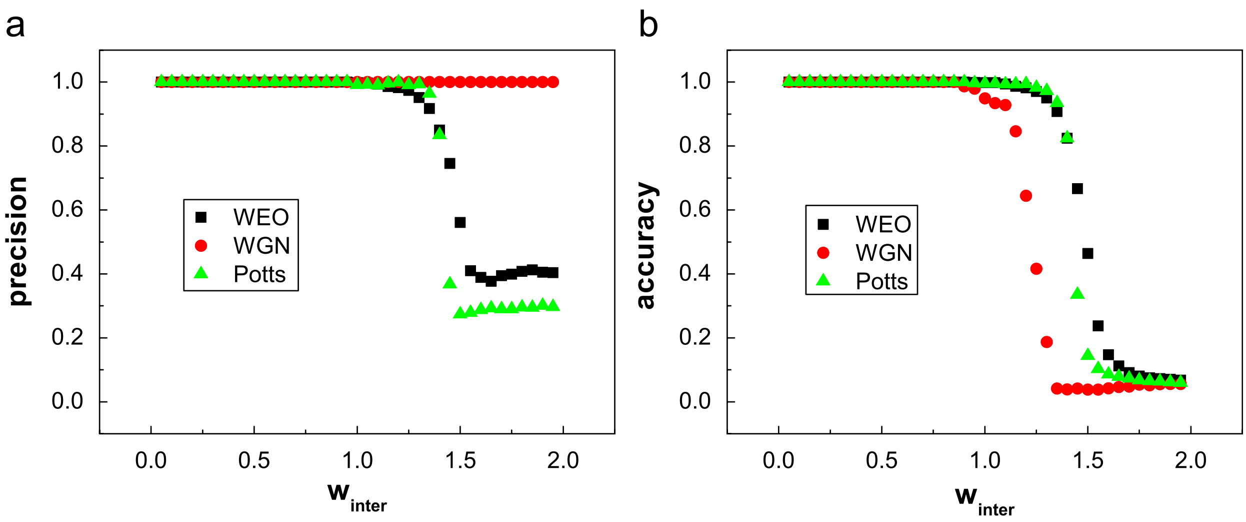

Fig. 3. Precision and accuracy of algorithms when $\langle k_{inter} \rangle$ is equal to 4. We see in the regime connection and weight are consistent($w_{inter} < 1$) all algorithms work well, but in the contradictory regime($w_{inter} > 1$), Potts and WEO perform better than WGN.

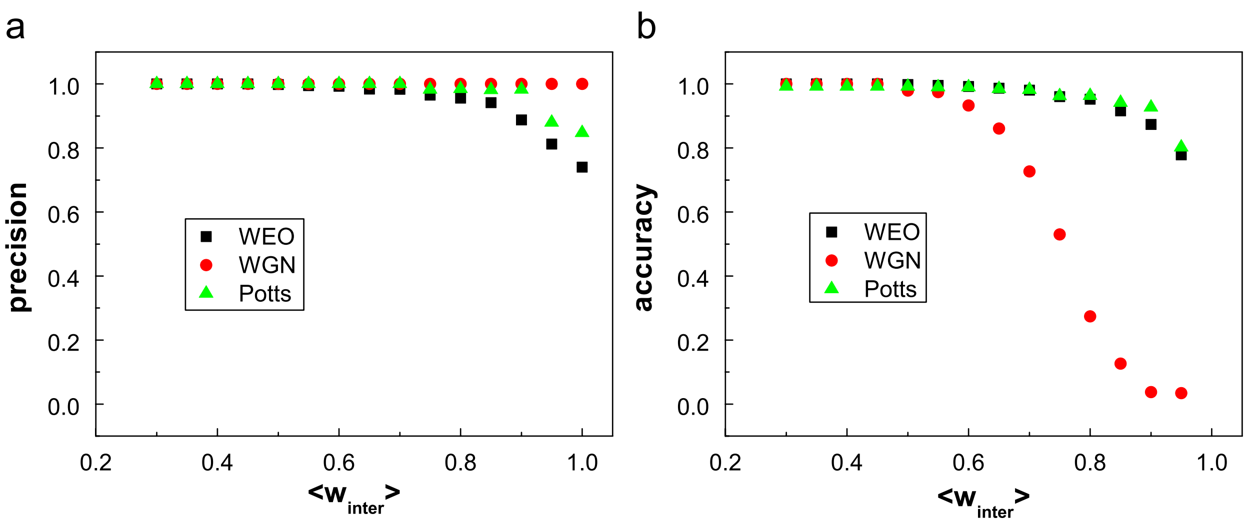

Fig. 4. The performance of algorithms with uniform distributed link weights. $\langle k_{inter} \rangle$ is equal to 8. The results are qualitatively similar with Fig. 2.

Although we focus on performance of algorithms, but a byproduct of this discussion is that we find that the precision of Potts/WEO algorithm may serve as an indicator of existence and distinctness of community structures in networks. Precision near 1 means there is distinct community structure and precision near 0 indicates there is no well-divided community structure.

2.4. Results on complete weighted networks

An extreme idealized example is the complete network. In complete networks, we use uniform distribution of link weights. Weights are taken randomly from the interval $[\langle w_{intra} \rangle - 0.25, \langle w_{intra} \rangle + 0.25]$ and $[\langle w_{inter} \rangle - 0.25, \langle w_{inter} \rangle + 0.25]$, respectively, for intragroup connections and intergroup connections. WGN algorithm works well when $\langle w_{inter} \rangle \ll \langle w_{intra} \rangle$, but its accuracy declines to almost zero when $\langle w_{inter} \rangle$ is greater than 0.5. While from Fig. 5, we can see that Potts and WEO algorithms perform quite well even for this extreme case.

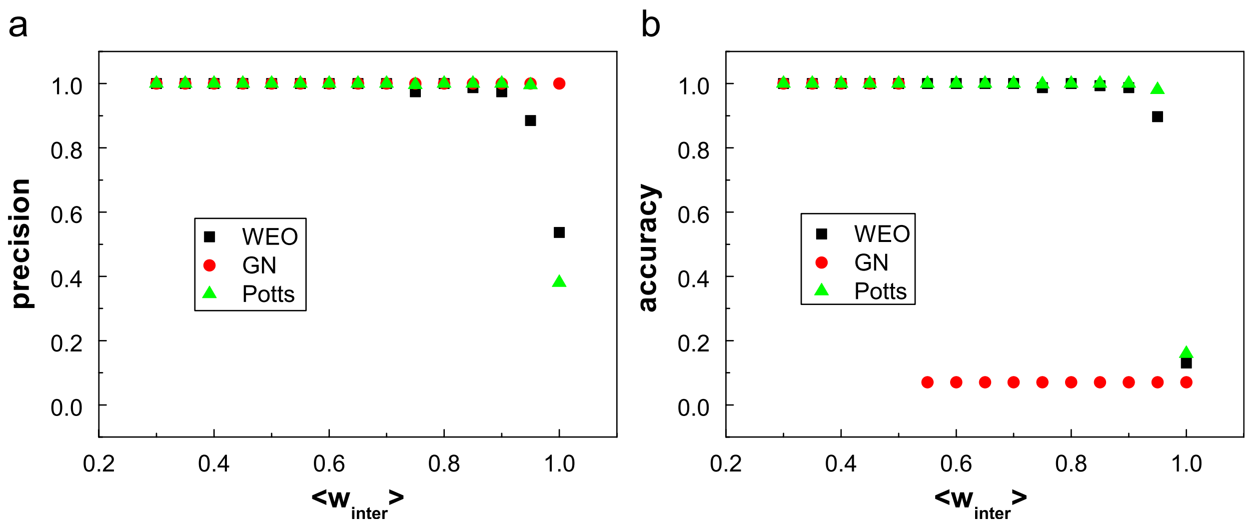

Fig. 5. Precision and accuracy of WGN, Potts and WEO algorithms in complete networks with presumed communities. When $\langle w_{inter}\rangle < \langle w_{intra}\rangle$, Potts and WEO algorithms both find the groups correctly.

3. Community structure in Rhesus monkey network

To investigate the performance of algorithms, we need also apply them to real weighted networks with known community structures. There are not so many real networks with reliable record of community structures. One example is Rhesus monkey network which is studied by Sade [35] in 1972. This network is based on observations of group F Cayo Santiago, in which 38 monkeys comprise 6 genealogies and 2 nonnatal males (066 and *R*006). The grooming episodes of monkeys were registered between 14th June and 31st July 1963, just prior to the mating season on Cayo Santiago. The network showed the information of members, who were 4 years old or older in 1963. Links denote grooming behavior between the monkeys and link weight w_{ij} is the total number of instances of grooming of each monkey by each other during the period of observation: $w_{ij} = w_{i\to j} + w_{j\to i}$, where $w_{i\to j}$ represents the number of i groomed j. The network has 16 vertices and 69 links with link weights ranging from 1 to 52. Its community structure revealed by WGN, Potts and WEO are presented as following in Fig. 6.

Newman has illuminated its community structure by weighted WGN algorithm (see Fig. 2(b) in Ref. [29], we have redrawn the figure as Fig. 6(a)). Here we apply Potts model and WEO algorithms in the Rhesus monkey network. Two algorithms have been applied 20 times to get the communities. Fig. 6(b) and (c) show the result gotten by Potts and WEO algorithm. The similarities between these three communities are: 0.65 (WEO vs. Potts), 0.22 (WEO vs. WGN), and 0.36 (Potts vs. WGN). Through more detailed comparison with the record of the known organization of these monkeys, it could be found that results gotten by WEO and Potts are in better consistence. The final weighted modularity of three groups have also demonstrated this conclusion.

	Precision	Modularity	Accuracy[a]
WGN	1.0	0.12	Not good
Potts	0.95	0.23	Good
WEO	1.0	0.244	The best

[a]According to our interpolation of the original record, see the following paragraph.

From the record by Sade [35], male 006 had been dominant in group since at least 1960. Male *R*006 had been solitary in 1962, and joined this group in early 1963. *R*006 replaced 006 as dominant male in the fall of 1963. Based on the investigation of cliques, the following details can be known. The dominant male 066 was

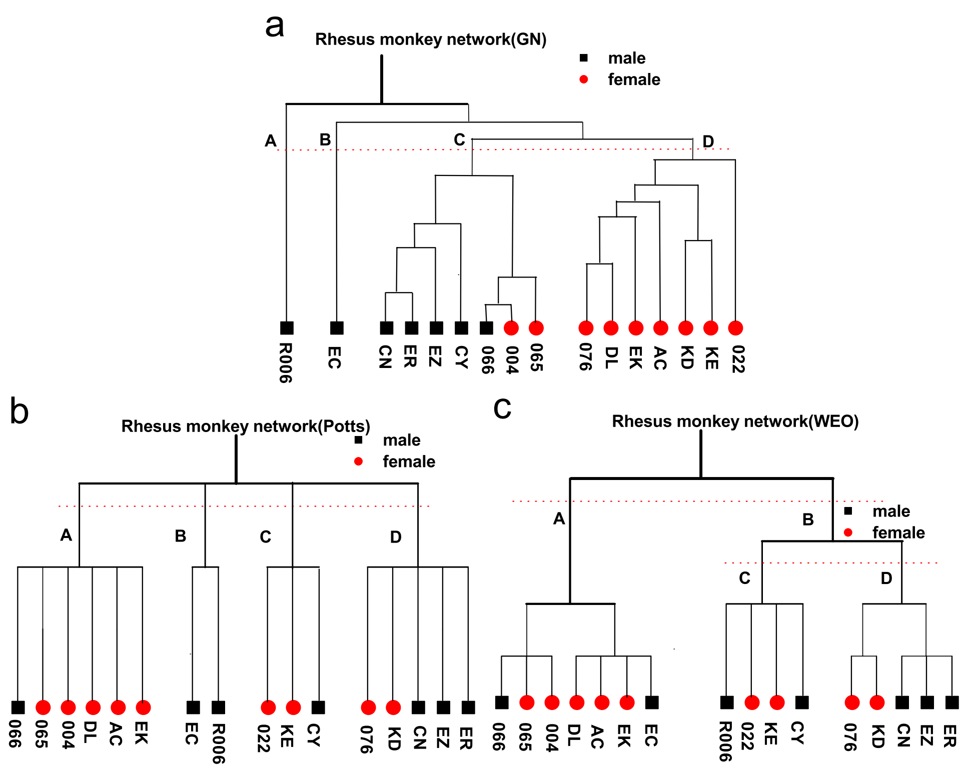

Fig. 6. (a) Community structure of Rhesus monkey network gotten by WGN algorithm given by Newman in Ref. [29]. (b) Result given by Potts algorithm. (c) Groups gotten by WEO algorithm. Those are the results appearing with maximum probability. *DL* and *EC* were *AC*'s offspring, *CY* and *KE* were 022's offspring, and *EZ* and *ER* were brothers.

co-cliqual with the first and second dominant females. *EC*, a 4-year-old male, is co-cliqual with his mother *AC* and sister *DL*. The 4 multi-cliqual monkeys, 065, 004, *AC* and *DL*, were the 4 highest ranking females and they formed the core of the grooming network. This can be reflected by community A of Fig. 6(b) and (c). *EZ* occurs only in his brother *ER*'s clique. *CN*, the adult male castrate, is co-cliqual with his mother and sister, overlapped more extensively with the cliques containing the other natal males (community D of Fig. 6 (b) and (c) can show this detail), and might link their clique to the main group. *R*006, the new-natal male, at last did not clique with the dominant and third ranking females, 065 and *AC* and *AC*'s son *EC*. This can be reflected by community C of Fig. 6(c). In conclusion, the dominant male 066, was integrated into the core with the females. The new male, *R*006, was distantly attached to the core of females. One sub-adult male, *EC*, was still integrated into his genealogy. The other natal males formed a distinct sub-group. *CN*, the castrate, was intermediate in his position, which overlapped that of the natal males and the female core. The community structure found by WEO algorithm can illuminate above details well. This example together with the above results on idealized networks suggest that WEO algorithm may be effective at finding community structure in weighted networks.

4. Conclusion

In this paper, using the ad hoc weighted networks and the Rhesus monkey network [35], we compare accuracy and precision of WGN, Potts and WEO algorithms. We find that when community structure is well-defined by topological linkage, all three algorithms work well. But for dense networks, when link weight plays more important rule in network properties, Potts and WEO algorithm work better. So WGN detects well-defined weighted and unweighted communities, and the other two algorithms work well in more dense networks (with or without weights), where communities are more fuzzy.

Another thing need to be noticed about this work is the entanglement between algorithm and network. Only when community structure of network is clear enough and well defined could we discuss the performance of algorithm, but community structure itself sometimes can only be discovered after we apply algorithms on the network. Therefore, when we find that the community structure detected by an algorithm is not stable, we do not know the unstableness ascribes the structure of network or instability of the algorithm. Fortunately, from above investigations we find that the precision of Potts/WEO algorithm could be used as an indicator of existence and distinctness of communities in networks. Therefore, if the precision from Potts/WEO is too low we will prefer to believe that the network itself has ambiguous community structure. This need to be confirmed further in the future.

Acknowledgment

The authors want to thank Dr. Newman for his cooperation data. This work is partially supported by 985 Project and NSFC under the Grant nos. 70431002, 70371072 and 70471080.

References

[1] M. Girvan, M.E.J. Newman, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.
[2] M. Boss, H. Elsinger, M. Summer, S. Thurner, arXiv:cond-mat/0309582 (2003).
[3] E. Ravasz, A.L. Somera, D.A. Mongru, Z.N. Olvai, A.L. Barabási, Science 297 (2002) 1551.
[4] R. Guimerà, L.A.N. Amaral, Nature 433 (2005) 895–900.
[5] P. Holme, M. Huss, H. Jeong, Bioinformatics 19 (2003) 532.
[6] P. Gleiser, L. Danon, Adv. Complex Syst. 6 (2003) 565–573.
[7] R.J. Williams, N.D. Martinez, Nature 404 (2000) 180–183.
[8] J.-P. Eckmann, E. Moses, Proc. Natl. Acad. Sci. 99 (2002) 5825.
[9] H. Zhou, R. Lipowsky, Lecture Notes Comput. Sci. 3038 (2004) 1062–1069.
[10] J.R. Tyler, D.M. Wilkinson, B.A. Huberman, arXiv:cond-mat/0303264 (2003).
[11] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, A. Arenas, Phys. Rev. E 68 (2003) 065103.
[12] M.E.J. Newman, M. Girvan, Phys. Rev. E 69 (2004) 026113.
[13] M.E.J. Newman, Phys. Rev. E. 69 (2004) 066133.
[14] M. Fiedler, Czech. Math. J. 23 (1973) 298;
A. Pothen, H. Simon, K.-P. Liou, SIAM J. Matrix Anal. Appl. 11 (1990) 430.
[15] B.W. Kernighan, S. Lin, Bell Syst. Technol. J. 49 (1970) 291.
[16] J. Scott, Sage, second ed., London, 2000.
[17] M. Young, J. Sager, G. Csárdi, P. Hága, arXiv:cond-mat/0408263 (2004).
[18] A. Capoccia, V.D.P. Servedioa, G. Caldarellia, F. Colaiori, Physica A 352 (2005) 669–676.
[19] S.-W. Son, H. Jeong, J.D. Noh, Eur. Phys. J. B 50 (2006) 431.
[20] F. Wu, B.A. Huberman, Eur. Phys. J. B 38 (2004) 331–338.
[21] L. Donetti, M.A. Muñoz, J. Stat. Mech. P (2004) 10012.
[22] H. Zhou, Phys. Rev. E 67 (2003) 061901.
[23] R. Guimerà, L.A.N. Amaral, J. Stat. Mech. (2005) P02001.
[24] J. Reichardt, S. Bornholdt, Phys. Rev. Lett. 93 (2004) 218701.
[25] J. Duch, A. Arenas, Phys. Rev. E 72 (2005) 027104.
[26] C.P. Massen, J.P.K. Doye, Phys. Rev. E 71 (2005) 046101.
[27] F. Wu, B.A. Huberman, Eur. Phys. J. B 38 (2004) 331–338.
[28] A. Medus, G. Acuña, C.O. Dorso, Physica A 358 (2005) 593–604.
[29] M.E.J. Newman, Phys. Rev. E 70 (2004) 056131.
[30] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, J. Stat. Mech. (2005) P09008.
[31] P. Zhang, M. Li, J. Wu, Z. Di, Y. Fan, Physica A 367 (2006) 577–585.
[32] L. Freeman, Sociometry 40 (1977) 35.
[33] S. Boettcher, A.G. Percus, Phys. Rev. Lett. 86 (2001) 5211.
[34] J. Reichardt, S. Bornholdt, Phys. Rev. E 74 (2006) 016110.
[35] D.S. Sade, Folia Primatol. 18 (1972) 196–223.