Detecting Community Structure of Complex Networks by Affinity Propagation

Jian Liu
LMAM and School of Mathematical
Sciences, Peking University
Beijing 100871, P.R. China
Email: dugujian1984@sina.com

Na Wang
Beijing University of Posts and
Telecommunications
Beijing 100876, P.R. China
Email: nawang007@163.com

Abstract—**The question of finding the community structure of a complex network has been addressed in many different ways. Here we utilize a clustering method called affinity propagation, associating with some existent measures on graphs, such as the shortest path, the diffusion distance and the dissimilarity index, to solve the network partitioning problem. This method considers all nodes as potential exemplars, and transmits real valued messages between nodes until a high quality set of exemplars and corresponding communities gradually emerges. It is demonstrated by simulation experiments that the algorithms can not only identify the community structure of a network, but also determine the number of communities automatically during the model selection. Moreover, they are successfully applied to several real-world networks, including the karate club network and the dolphins network.**

Keywords-**complex networks; community structure; affinity propagation; shortest path; diffusion distance; dissimilarity index.**

I. Introduction

There has been an explosive growth of interest and activity on the structure and dynamics of complex networks [1], [2], [3] during recent years. Network models have also become popular tools in social science, economics, the design of transportation and communication systems, banking systems, etc, due to our increased capability of analyzing these models. Since these networks are typically very complicated, we would like to find the best partition of this network into a small number of communities, which is a common feature of many networks [4], [5], [6], [7]. Community structure has the tendency for nodes to divide into clusters with dense connections within clusters and only sparser connections between them. A huge variety of community detection techniques have been developed during recent years [4], [5], [6], [7], which are constructed from different viewing angles comparing different proposals in the literature.

In [8], [9] Frey and Dueck devised a method called affinity propagation, which we utilize here to address the problem of finding community structure in networks. Affinity propagation method simultaneously considers all nodes as potential exemplars, then messages are exchanged between nodes until a good set of exemplars and corresponding communities gradually emerges. As described later, messages are updated on the basis of simple formulas that search for minima of an appropriately chosen energy function. At any point in time, the magnitude of each message reflects the current affinity that one node has for choosing another node as its exemplar.

The most widely known measure on a graph is the shortest path [4], [10], which denotes the minimum number of edges traversed to get form one node to another. The concept of diffusion distance was proposed in [6], where the main idea was to define a system of coordinates with an explicit metric and the construction was based on a Markov random walk on networks [11]. The diffusion distance between two nodes will be small if they are connected by many paths in the network. Another work [12], [13] was also along the lines of random walker Markovian dynamics, where a dissimilarity index based on the mean first passage time of the networks was conducted. As described later, the dissimilarity index between two nodes will be small if they belong to the same community and large if they belong to different communities.

We constructed the following algorithms—affinity propagation with the shortest path (APSP), affinity propagation with the diffusion distance (APDD) and affinity propagation with the dissimilarity index (APDI), via selecting the above measures on graphs. The algorithms have been applied to some artificial networks, including the sample network generated from Gaussian mixture model and the ad-hoc network, and also some real-world networks, including the social interactions between members of a karate club and the frequent associations between dolphins in a community living off Doubtful Sound, New Zealand. From the numerical performance we can see all of them succeed in small size networks while APDI has the highest efficiency in larger networks.

The rest of the paper is organized as follows. In the sequel, we first briefly review the basic idea of the affinity propagation method [8], then introduce the three selected measures on networks, including the shortest path, the diffusion distance and the dissimilarity index, and the algorithms are described in detail after these introductions in Section II. In Section III, we apply the algorithms to some model problems and real-world networks mentioned before. The clustering results are typically compared. Finally we make the conclusion in Section IV.

978-1-4244-4738-1/09/$25.00 ©2009 IEEE

II. Affinity propagation method for detecting community structure of networks

A. The Basic Idea of Affinity Propagation

In [8], [9] Frey and Dueck devised a method called affinity propagation, which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges.

Let $G(S,E)$ be a network with n nodes and m edges, where S is the nodes set, $E=\{e(x,y)\}_{x,y\in S}$ is the weight matrix and $e(x,y)$ is the weight for the edge connecting the nodes x and y. We take a partition of S as $S=\bigcup_{i=1}^{N} S_i$ with $S_i \bigcap S_j = \emptyset$ if $i \neq j$, then identifying exemplars can be viewed as searching over valid partitions $\mathbb{S}=\{S_1,\cdots,S_N\}$ so as to minimize the energy

$$E(\mathbb{S}) = -\sum_{i=1}^{N} s(x,m_i), \tag{1}$$

where $s(x,y)$, which is called similarity, indicates how well the node x is suited to node y, and m_i denotes the exemplar of S_i. Messages are updated on the basis of formulas that search for minima of (1). At any point in time, the magnitude of each message reflects the current affinity that one node has for choosing another node as its exemplar. When the points in Euclid space are considered, $s(x,y)=-\|x-y\|^2$. Indeed, the method described here can be applied when the optimization criterion is much more general [8]. Three kinds of measures based on graphs, which we will utilize to our partition problem, are demonstrated as shortest path, diffusion distance and dissimilarity index, as described later in Section II-B.

There are two kinds of messages exchanged between nodes, and each takes into account a different kind of competition. Messages can be combined at any stage to decide which exemplar it belongs to. The responsibility $r(x,y)$, sent from node x to candidate exemplar node y, reflects the accumulated evidence for how well-suited y is to serve as the exemplar for x, taking into account other potential exemplars for x. The availability $a(x,y)$, sent from candidate exemplar node y to node x, reflects the accumulated evidence for how appropriate it would be for x to choose y as its exemplar, taking into account the support from other points that y should be an exemplar. The responsibilities are computed using the rule

$$r(x,y) = s(x,y) - \max_{z\neq y}\Big\{a(x,z)+s(x,z)\Big\}. \tag{2}$$

Such responsibility update makes all candidate exemplars compete for ownership of a node, the following availability update gathers evidence from nodes as to whether each candidate exemplar would make a good exemplar

$$a(x,y) = \min\Big\{0, r(y,y) + \sum_{z\neq x,y}\max\big\{0, r(z,y)\big\}\Big\}. \tag{3}$$

The self-availability $a(y,y)$ is updated differently

$$a(y,y) = \sum_{z\neq y}\max\Big\{0, r(z,y)\Big\}. \tag{4}$$

This message reflects accumulated evidence that node y is an exemplar, based on the positive responsibilities sent to candidate exemplar y from other nodes.

The above update rules require some computations we omit here, which were implemented in detail in [8], [9]. Responsibilities and availabilities can be combined to identify exemplars during affinity propagation. For node x, the node y that maximizes $r(x,y)+a(x,y)$ either identifies x as an exemplar if $y=x$, or identifies the node that is the exemplar for x. The message passing procedure may be terminated after L iterations after changes in the messages fall below a threshold. When updating the messages, it is important that they are damped to avoid numerical oscillations that arise in some circumstances. Each message is set to λ times its value from the previous iteration plus $1-\lambda$ times its prescribed updated value, where $\lambda\in[0,1]$.

Affinity propagation can be viewed as a method that searches for minima of the energy function (1) depends on a set of N hidden labels $m_1,\cdots,m_N$, corresponding to the N nodes. Each label indicates the exemplar to which the node belongs. Exactly minimizing the energy is computationally intractable, since a special case of this minimization problem is the NP-hard k-median problem [14]. However, the update rules for affinity propagation correspond to fixed-point recursions for minimizing a Bethe free-energy [15]. Affinity propagation is most easily derived as an instance of the max-sum algorithm in a factor graph [16] describing the constraints on the labels and the energy functions.

B. Similarity Between Nodes in Networks

1) Shortest Path: The concept of the shortest path in graph theory has been frequently applied in networks [4], [10]. Let $S(x,y)$ be the shortest path between node x and node y, which denotes the minimum number of edges traversed to get form x to y. There may not be an unique shortest path between two nodes. The shortest paths between any two nodes x and y in the network can be calculated using the following procedure [10]

(1) Assign node x distance zero, to indicate that it is zero steps away from itself, and set $d=0$.
(2) For each node z whose assigned distance is d, follow each attached edge to the node w at its other end and, if w has not already been assigned a distance, assign it distance $d+1$. Declare z to be a predecessor of w.
(3) If w has already been assigned distance $d+1$, then there is no need to do this again, but z is still declared a predecessor of w.
(4) Set $d=d+1$.
(5) Repeat from step 2 until there are no unassigned vertices left.

Now the shortest path from x to y is the path that get by stepping from x to its predecessor, and then to the predecessor of each successive node until y is reached. If a node has two or more predecessors, then there are two or more shortest paths, each of which must be followed separately if we wish to know all shortest paths from x to y.

2) Diffusion Distance: The main idea of [6] is to define a system of coordinates with an explicit metric that reflects the connectivity of nodes in a given network and the construction is based on a Markov random walk on networks. The network can be related to a discrete-time Markov chain with stochastic matrix P with entries $p(x,y)$ given by

$$p(x,y)=\frac{e(x,y)}{d(x)}, \qquad d(x)=\sum_{z\in S}e(x,z), \tag{5}$$

where $d(x)$ is the degree of the node x [11], [17]. This Markov chain has stationary distribution

$$\mu(x)=\frac{d(x)}{\sum_{z\in S}d(z)} \tag{6}$$

and it satisfies the detailed balance condition [7]. The diffusion distance $D(x,y)$ between x and y is defined as the weighted L^2 distance

$$D(x,y)=\left(\sum_{z\in S}\frac{\Big(p(x,z)-p(y,z)\Big)^2}{\mu(z)}\right)^{\frac{1}{2}}, \tag{7}$$

where the weight $\mu(z)^{-1}$ penalize discrepancies on domains of low density more than those of high density.

The transition matrix P has a set of eigenvectors and eigenvalues $1=\lambda_0\geq|\lambda_1|\geq\cdots\geq|\lambda_{n-1}|\geq 0$

$$P\varphi_k=\lambda_k\varphi_k, \quad k=0,1,\cdots,n-1. \tag{8}$$

Let K be the largest index k such that $|\lambda_k|>\delta|\lambda_1|$ and if we introduce the diffusion map

$$\Psi: x\longmapsto\begin{pmatrix}\lambda_1\varphi_1(x)\\ \vdots\\ \lambda_K\varphi_K(x)\end{pmatrix}, \tag{9}$$

then the diffusion distance $D(x,y)$ can be approximated to relative precision δ using the first K non-trivial eigenvectors and eigenvalues

$$D(x,y)\simeq\sum_{k=1}^{K}\lambda_k^2\Big(\varphi_k(x)-\varphi_k(y)\Big)^2=\|\Psi(x)-\Psi(y)\|^2. \tag{10}$$

This notion of similarity of nodes in the network reflects the intrinsic geometry of the set in terms of connectivity of the data points in a diffusion process. The diffusion distance between two nodes will be small if they are connected by many paths in the network. This metric is thus a key quantity in the design of inference algorithms that are based on the preponderance of evidences for a given hypothesis. Furthermore, it is usually more appropriate than the shortest path when propagating the information from a labeled example x to the new point y, as it takes into account all paths relating x to y.

3) Dissimilarity Index: In [12], a Brownian particle is introduced into a network to measure the distances between nodes and a quantity called dissimilarity index between pairs of nodes is defined, which signifies to what extent two nodes would like to be in the same community. Suppose the Brownian particle is located at node x. The mean first passage time $t(x,y)$ is the average number of steps it takes before it reaches node y for the first time, which is given by

$$\begin{aligned}t(x,y)= \quad & p(x,y)+\sum_{m=1}^{+\infty}(m+1)\\ & \cdot\sum_{z_1,\cdots,z_m\neq y}p(x,z_1)p(z_1,z_2)\cdots p(z_m,y).\end{aligned} \tag{11}$$

It has been shown that $t(x,y)$ is the solution of the linear equation [12]

$$[I-B(y)]\begin{pmatrix}t(1,y)\\ \vdots\\ t(n,y)\end{pmatrix}=\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix} \tag{12}$$

where $B(y)$ is the matrix formed by replacing the y-th column of matrix P with a column of zeros. Taking any node x as the origin of the network , then the set $\{t(x,z): z\in S, z\neq x\}$ measures how far all the other nodes are located from the origin x. Suppose node x and y are nearest neighbors, the difference between them can be quantitatively measured. The dissimilarity index is defined by the following expression

$$I(x,y)=\frac{1}{n-2}\left(\sum_{z\in S,z\neq x,y}\Big(t(x,z)-t(y,z)\Big)^2\right)^{\frac{1}{2}}. \tag{13}$$

If two nearest neighboring nodes x and y belong to the same community, then the average distance $t(x,z)$ from x to any other node z will be quite similar to the average distance $t(y,z)$ from y to z. Consequently, $I(x,y)$ in (13) will be small if x and y belong to the same community and large if they belong to different communities.

C. The Algorithms

We have selected three kinds of measures on the graph as the similarities between nodes in the network. Based on the ideas described above, the affinity propagation (AP) algorithm for detecting community structure of networks works as follows

(1) Initialize messages with responsibilities and availabilities setting to be zero; Set damping factor λ and the number of iterations R.

(2) Choose any kind of measure in Section II-B as similarity $\{s(x,y)\}_{x,y\in S}$.

(3) For $l=0,1,\cdots,L$, do the following

  (3.1) Compute responsibilities $R^{(l+1)}$ using (2), then dampen it by $R^{(l+1)}=(1-\lambda)\cdot R^{(l+1)}+\lambda\cdot R^{(l)}$;

  (3.2) Compute availabilities $A^{(l+1)}$ using (3) and (4), then dampen it by $A^{(l+1)}=(1-\lambda)\cdot A^{(l+1)}+\lambda\cdot A^{(l)}$.

(4) The final responsibilities and availabilities are combined to identify exemplars. Let $C=R+A$, then for each node

x, we obtain the exemplar z of x by

$$z = \arg\max_{y \in S} C(x, y), \tag{14}$$

which can either identifies node x as an exemplar if $z = x$, or identifies node z as the exemplar for node x.

According to different similarities, we obtain three forms of algorithms—affinity propagation with the shortest path (APSP), affinity propagation with the diffusion distance (APDD) and affinity propagation with the dissimilarity index (APDI), by which we denote them later.

The computational cost of affinity propagation algorithm is described as follows. It is easy to see that the computational costs of responsibilities and availabilities in each iteration are both $O(n^2)$. Let us estimate the cost for the three similarities.

- Shortest path. In the standard implementation of this procedure [10], a queue is maintained of nodes whose distances have been assigned, but whose attached edges have not yet been followed. This allows the procedure to run to completion in time $O(m)$, where m is the number of edges in the graph. We note also that the procedure allows us to calculate the shortest paths from all vertices to the target y in a single run, and not just from the single node x that we were originally interested in. Thus the calculation of n shortest paths is in time $O(m)$. The total cost for the shortest paths in a network with n nodes and m edges is $O(mn)$.
- Diffusion distance. For a given network and precision, the diffusion map Ψ in (9) is a matrix of order $K \times n$. It is easy to find that the computation of diffusion distance (10) is $O(Kn^2)$.
- Dissimilarity index. Note that (13) seems to imply that matrix inversion operations are needed to calculate the values of $t(x, y)$ for all pairs x and y. This would lead to a computational time of $O(n^4)$. However, since what we really need to know is the difference of mean first passage times, i.e. $t(x, z) - t(y, z)$, we utilize the strategy described by which one can calculate all the different differences with a computational time of $O(n^3)$ [13].

Although the measure of diffusion distance and dissimilarity index cost more than the shortest path, it can lead to a more reasonable partitioning result according to our practical experiments.

III. Experimental results

A. Sample Networks Generated from Gaussian Mixture Model

To test the validity of the algorithms, we apply them to a sample network generated from a Gaussian mixture model. This model is closely related the concept random geometric graph proposed by Penrose [18] except that we take Gaussian mixture here compared with uniform distribution in [18].

The procedure is operated as the following. Firstly, we generate n sample points $\{\boldsymbol{x}_i\}$ in two dimensional Euclidean space subject to a K-Gaussian mixture distribution

$$\sum_{i=1}^{K} q_i G\left(\boldsymbol{\mu}_i, \Sigma_i\right), \tag{15}$$

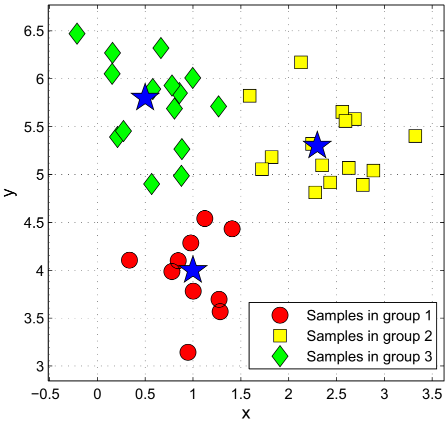


Fig. 1. 40 sample points generated from the given 3-Gaussian mixture distribution. The star symbols represent the centers of each Gaussian component. The circle, square and diamond shaped symbols represent the position of sample points in each component respectively.

where $\{q_i\}$ are mixture proportions satisfying $0 < q_i < 1$, $\sum_{i=1}^{K} q_i = 1$. $\boldsymbol{\mu}_i$ and Σ_i are the mean positions and covariance matrices for each component, respectively. Then we generate the network with a thresholding strategy. That is, if $|\boldsymbol{x}_i - \boldsymbol{x}_j| \leq dist$, we set an edge between the i-th and j-th node; otherwise they are not connected. With this strategy, the topology of the network is induced by the metric. As a consequence, some properties of the network, say the clustering nature, may be inherited from the case with metric. This is our basic motivation with this model.

First, we take $n = 40$ and $K = 3$, then generate the sample points with the means

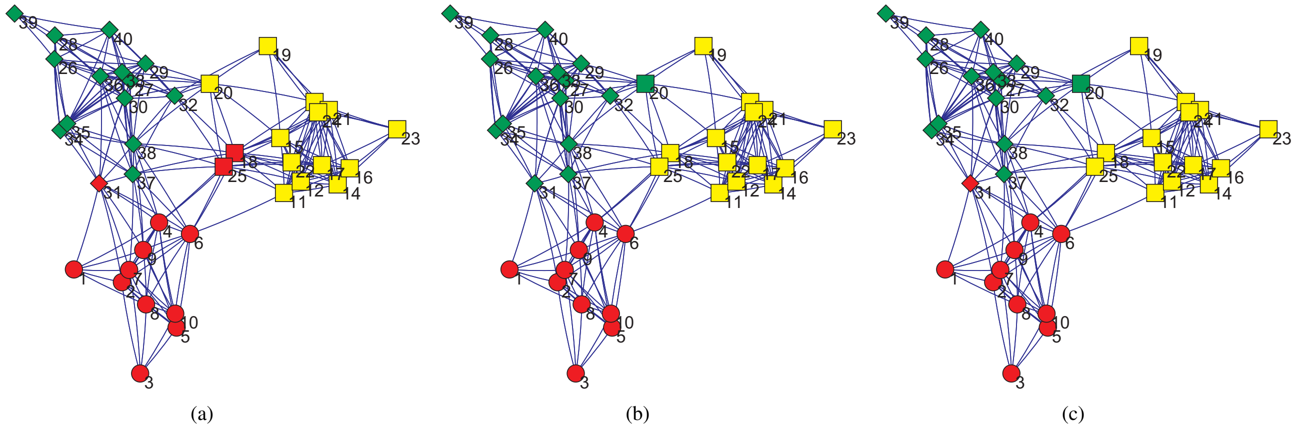
$$\boldsymbol{\mu}_1 = (1.0, 4.0)^T, \boldsymbol{\mu}_2 = (2.3, 5.3)^T, \boldsymbol{\mu}_3 = (0.5, 5.8)^T, \tag{16a}$$

and the covariance matrices

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.15 \end{pmatrix}. \tag{16b}$$

Here we pick nodes 1:10 in group 1, nodes 11:25 in group 2 and nodes 26:40 in group 3 for simplicity. With this choice, approximately $q_1 = 10/40, q_2 = q_3 = 15/40$. The sample points are shown in Figure 1. We take $dist = 1.0$ and then generate the network. By setting $\lambda = 0.5$ and $R = 100$, we make clustering with our algorithms. The results are shown in Figure 2. It seems all of three cases perform successfully, and all of them can obtain the number of communities $N = 3$, which is consistent with our motivation of this model $K = 3$. The results of APDD and APDI are more reasonable, of which the classification coincide except the node 31. Node 18, 20, 25 appear to be unreasonably clustered in APSP, since the measure of shortest path sometimes can not reflect cohesion of a community, but only the connection between two nodes in networks.

Next we take $n = 320$ and $K = 4$, where nodes 1:80 are in group 1, nodes 81:160 in group 2, nodes 161:240 in group



(a)

(b)

(c)

Fig. 2. Partition of the network generated from the sample points in Figure 1 with the parameter $dist = 1.0$. The three cases perform successfully. (a)APSP; (b)APDD; (c)APDI.

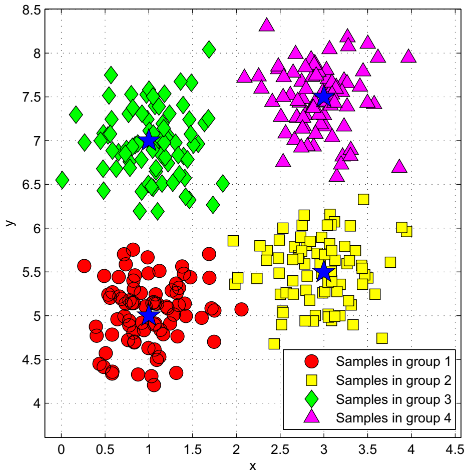


Fig. 3. 320 sample points generated from the given 4-Gaussian mixture distribution. The star symbols represent the centers of each Gaussian component. The circle, square, diamond and triangle shaped symbols represent the position of sample points in each component respectively.

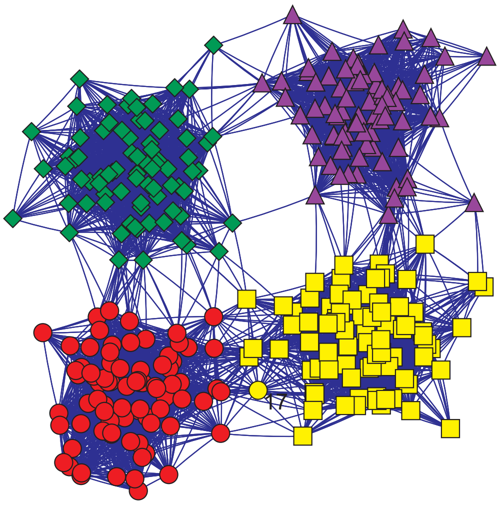


Fig. 4. Partition of the network generated from the sample points in Figure 3 with the parameter $dist = 0.8$ by using APDI. The community structure coincide with the original sample model expect node 17.

3 and nodes 241:320 in group 4. This means approximately $q_1 = q_2 = q_3 = q_4 = 80/320$. The other model parameters are chosen as

$$\begin{aligned}\boldsymbol{\mu}_1 = (1.0, 5.0)^T, \boldsymbol{\mu}_2 = (3.0, 5.5)^T,\\ \boldsymbol{\mu}_3 = (1.0, 7.0)^T, \boldsymbol{\mu}_4 = (3.0, 7.5)^T,\end{aligned} \tag{17a}$$

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.15 \end{pmatrix}. \tag{17b}$$

Here we take $dist = 0.8$ and set $\lambda = 0.5$ and $R = 1000$. The sample points are shown in Figure 3 and the computational result of APDI is shown in Figure 4 where we obtain $N = 4$. It indicates that APDI goes smoothly with several hundreds of nodes, while the other two can not obtain a reasonable result intuitively.

B. Ad hoc networks with 128 nodes

We apply our methods to the ad-hoc network with 128 nodes in this subsection. The ad-hoc network is a typical benchmark problem considered in many papers [4], [7], [19]. It has a known community structure and is constructed as follows. Suppose we choose $n = 128$ nodes, split into 4 communities containing 32 nodes each. Assume that pairs of nodes belonging to the same communities are linked with probability p_{in}, and pairs belonging to different communities with probability p_{out}. These values are chosen so that the average node degree, d, is fixed at $d = 16$. In other words p_{in} and p_{out} are related as

$$31p_{\text{in}} + 96p_{\text{out}} = 16. \tag{18}$$

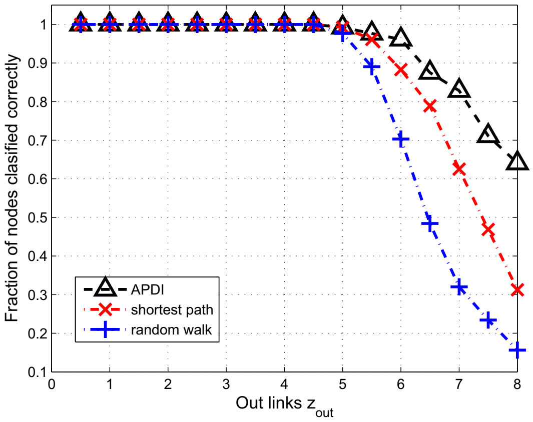


Fig. 5. The fraction of nodes classified correctly by APDI and the methods used in [4]. It seems that APDI has better partition result than shortest path and random walk methods [4].

Here we naturally choose the nodes group $S_1 = \{1 : 32\}, S_2 = \{33 : 64\}, S_3 = \{65 : 96\}, S_4 = \{97 : 128\}$.

We change z_{out} from 0.5 to 8 and look into the fraction of nodes which correctly classified. By setting $\lambda = 0.5$, $R = 100$, we make clustering by APDI. The fraction of correctly identified nodes is shown in Figure 5, comparing with the two methods described in [4]. APDI perform noticeably better than the two previous methods, especially for the more difficult cases when z_{out} is large.

C. The karate club network

This network was constructed by Wayne Zachary after he observed social interactions between members of a karate club at an American university [20]. Soon after, a dispute arose between the club's administrator and main teacher, then the club split into two smaller clubs. It is used in several papers to test the community structure algorithms [4], [7]. There are only 34 nodes in karate club network. By setting $\lambda = 0.5$ and $R = 100$, we find the results of the algorithms APSP, APDD and APDI, exit some differences, which are shown in Figure 6. It seems that SASP and SADI obtain a better result in this case. The confused community structure resulted by SADD even reflects connectivity of the nodes in a diffusion process.

D. The dolphins network

The dolphins network is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand [21]. The network was compiled from the studies of the dolphins, with ties between dolphin pairs being established by observation of statistically significant frequent association. By setting $\lambda = 0.5$ and $R = 100$, we obtain the partitioning results shown in Figure 7. According to the results, the network seems splitting into two large communities by the green part and the larger one at first, then the larger of the two keep splitting into a few smaller communities, represent by different colors. The split into two groups appears to correspond to a known division of the dolphin community [22]. The subgroupings within the larger half of the network also seem to correspond to real divisions among the animals that the red part in Figure 7(b) consists almost of entirely of females and the others almost entirely of males, which is more similar to the result obtained by SADD.

IV. Conclusions

We utilize affinity propagation algorithm with three measures on graphs as the input similarities to address the network partitioning problem in this paper. This method considers all nodes as potential exemplars, and transmits real valued messages between nodes until a high quality set of exemplars and corresponding communities gradually emerges. The algorithms—SADD, SADI and SASP, are proposed and successfully applied to some artificial networks. The numerical performance have shown that all of them can give reasonable partitioning results, while we recommend APDI here for its accuracy in most cases. Moreover, the algorithms succeed in two real-world learning tasks, including the karate club network and the dolphins network. We again point out that our algorithms can not only identify the community structure of a network, but also automatically determine the number of communities during the model selection, which people are sometimes interested in [4], [19].

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant 10871010 and the National Basic Research Program of China under grant 2005CB321704. The authors thank Prof. Tiejun Li for many discussions.

References

[1] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, no. 1, pp. 47–97, 2002.

[2] M. Newman, A.-L. Barabási, and D. J. Watts, *The structure and dynamics of networks*. Princeton: Princeton University Press, 2005.

[3] N. R. Council, *Network Science. National Academy of Sciences*, Washington DC, 2005.

[4] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, 2004.

[5] M. Newman, "Detecting community structure in networks," *Eur. Phys. J. B*, vol. 38, no. 2, pp. 321–330, 2004.

[6] S. Lafon and A. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. Pattern. Anal. Mach. Intel.*, pp. 1393–1403, 2006.

[7] W. E, T. Li, and E. Vanden-Eijnden, "Optimal partition and effective dynamics of complex networks," *Proc. Natl. Acad. Sci. USA*, vol. 105, no. 23, pp. 7907–7912, 2008.

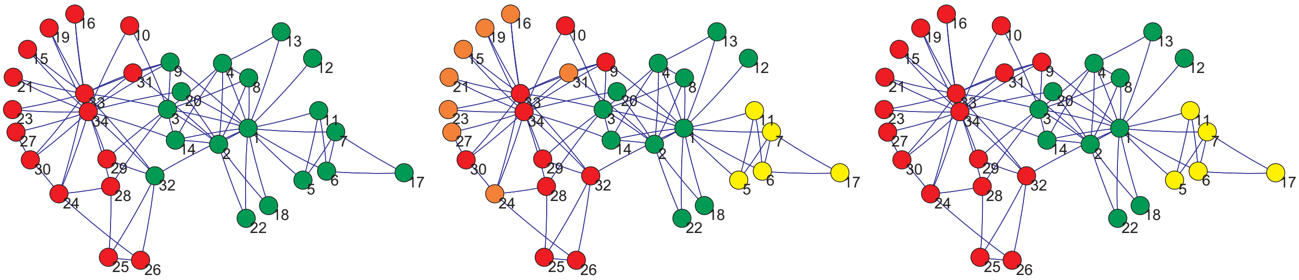
[8] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, p. 972, 2007.

[9] ——. (2007) Supporting online material of clustering by passing messages between data points. [Online]. Available: http://www.sciencemag.org/cgi/content/full/1136800/DC1

[10] M. Newman, "Scientific collaboration networks:ii.shortest paths, weighted networks, and centrality," *Phys. Rev. E*, vol. 64, no. 1, p. 016132, 2001.

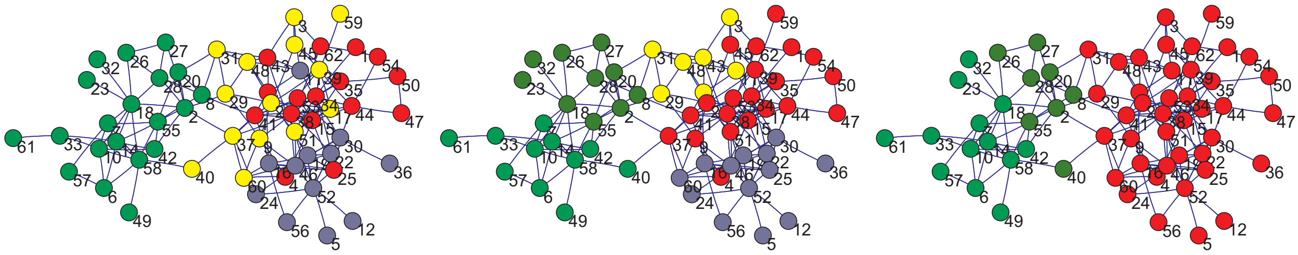
[11] L. Lovasz, "Random walks on graphs: A survey," *Combinatorics, Paul Erdos is Eighty*, vol. 2, pp. 1–46, 1993.

[12] H. Zhou, "Distance, dissimilarity index, and network community structure," *Phys. Rev. E*, vol. 67, no. 6, p. 061901, 2003.



(a) (b) (c)

Fig. 6. The community structure of the karate club network [20] using our algorithms. (a)APSP; (b)APDD; (c)APDI.



(a) (b) (c)

Fig. 7. The community structure of the dolphins network [21], [22] using our algorithms. (a)APSP; (b)APDD; (c)APDI.

[13] H. Zhou and R. Lipowsky, "Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities," *Lecture Notes in Computer Science*, vol. 3038, pp. 1062–1069, 2004.

[14] M. Charikar, S. Guha, É. Tardos, and D. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *J. Comput. Syst. Sci.*, vol. 65, no. 1, pp. 129–149, 2002.

[15] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

[16] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[17] F. Chung, *Spectral Graph Theory. American Mathematical Society*, Rhode Island, 1997.

[18] M. Penrose, *Random Geometric Graphs*. Oxford: Oxford University Press, 2003.

[19] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech.*, vol. 9, p. P09008, 2005.

[20] W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthrop. Res.*, vol. 33, no. 4, pp. 452–473, 1977.

[21] D. Lusseau, "The emergent properties of a dolphin social network," *Proceedings of the Royal Society B: Biological Sciences*, vol. 270, pp. 186–188, 2003.

[22] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.