Research Article

Incremental Density-Based Link Clustering Algorithm for Community Detection in Dynamic Networks

Fanrong Meng,¹ Feng Zhang,¹ Mu Zhu,^{1,2} Yan Xing,¹ Zhixiao Wang,¹ and Jihong Shi^{1,3}

¹School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

²State Key Laboratory of NBC Protection for Civilian, Research Institute of Chemical Defense, Beijing 102205, China

³Zhengzhou Coal Industry (Group) Limited Liability Company, Zhengzhou 450000, China

Correspondence should be addressed to Feng Zhang; fern1991@163.com

Received 24 September 2015; Revised 16 December 2015; Accepted 20 December 2015

Academic Editor: Juan A. Almendral

Copyright © 2016 Fanrong Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Community detection in complex networks has become a research hotspot in recent years. However, most of the existing community detection algorithms are designed for the static networks; namely, the connections between the nodes are invariable. In this paper, we propose an incremental density-based link clustering algorithm for community detection in dynamic networks, iDBLINK. This algorithm is an extended version of DBLINK which is proposed in our previous work. It can update the local link community structure in the current moment through the change of similarity between the edges at the adjacent moments, which includes the creation, growth, merging, deletion, contraction, and division of link communities. Extensive experimental results demonstrate that iDBLINK not only has a great time efficiency, but also maintains a high quality community detection performance when the network topology is changing.

1. Introduction

In the real world, many systems can be abstracted as a network, such as the Internet, interpersonal relationship networks, disease transmission networks, and scientist cooperation networks. A large number of studies have shown that community structure exists in complex networks. Nodes are connected closely within the community, while the connections between the communities are relatively sparse. Community structure plays an increasing important role in the complex networks. It can help us understand the function of the complex networks, find the potential law, and predict the behavior of complex networks [1]. For example, community structure represents the group with the same interests and hobbies in the social networks; in the field of Web, Web pages have more links within the same community; in the literature network, literatures within the same community have related research topic. In a word, it has important theoretical significance and practical value to study the community structure of complex networks.

At present, great progress has been achieved in the research on community detection of complex networks and

many representative algorithms have been put forward. Hierarchy centered algorithms use the idea of hierarchical clustering, so that the original complex networks are built into a hierarchical community structure. These algorithms include clustering and classification methods, whose representative algorithms are GN [2] and P-SNCD [3]. Node centered algorithms make strict demands on every node of the complex networks to carry on community detection which is represented by CPM [4] algorithm. Link centered algorithms include LINK [5] and DBLINK [6]; DBLINK algorithm is proposed in our previous work. The framework in [7] gives the basis for multidimensional network analysis, and the method in [8] also uses a partition of the links of a network in order to uncover its community structure. Group centered algorithms define a community of network and find all the communities which meet this definition in the network. There are many typical algorithms, such as LFM [9], DOCA [10], and OSLOM [11]. Network centered algorithms use certain standard (such as modularity [12]) to divide the entire network, and, by finding the best partition to represent the community structure, CNM [13] and AdClust [14] are good examples. Other algorithms include label propagation

algorithms [15, 16], algorithms based on information dissemination [17, 18], and so on.

Most existing community detection algorithms are aimed at static networks; namely, the connections between the nodes in the network are invariable. However, networks are not the case in many practical applications which are dynamic. Taking an example of microblog and other social networks, from a macro perspective, the number of users will continue to grow, making the network scale increase. And from the microscopic point of view the users can add new friends or delete existing good friend relationship. These changes will result in changes of the entire network community structure, such as the formation of new communities, the disappearance of old communities, and the merger or division of communities, as well as the larger or smaller community size. Although LabelRankT [19], QCA [20], DENGGRAPH [21], AFOCS [22], and other incremental community algorithms have been proposed in succession, community detection in dynamic networks still faces enormous challenges due to the rapid and unpredictable changes.

This paper proposes an incremental density-based link clustering algorithm for community detection in dynamic networks, iDBLINK, which in essence is an extension of DBLINK algorithm. It can update the current affected local community structure according to different network topology changes. The advantages of iDBLINK mainly embody in the following aspects:

- (1) iDBLINK can update the link community structure in the current network quickly and efficiently through the intensity changes of similarity relation between the edges and link community structure a moment before.
- (2) iDBLINK has inherited the advantage of DBLINK algorithm which can effectively identify overlapping community structure. It only needs to consider the changes of the disjoint link communities, while the overlapping node communities can be naturally reflected according to the link communities.
- (3) Experimental results in the simulative and real networks show that DBLINK can find overlapping community structure with higher quality compared with some other representative community detection algorithms, while iDBLINK algorithm can be better applied to the dynamic network environment.

2. Static Network Community Detection DBLINK

Usually, a complex network can be shown by a graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ represents node set and $E = \{e_1, e_2, \dots, e_m\}$ represents edge set; n and m , respectively, are the number of nodes and edges in a network. Any edge $e = (u, v)$ means the two nodes connecting e are u and v .

Definition 1. If two edges share a common node, then there is similarity between them. The similarity between Edge $e = (u, v)$ and edge $l = (u, w)$ is defined as

$$S(e, l) = \frac{|\Gamma(v) \cap \Gamma(w)|}{|\Gamma(v) \cup \Gamma(w)|}. \quad (1)$$

$\Gamma(v)$ is the node set consisting of the node v itself and all its neighbor nodes, so $\Gamma(v) = \{x \mid (v, x) \in E\} \cup \{v\}$.

Definition 2. $N(e)$ is the neighbor of $e = (u, v)$ which refers to a collection of edges, respectively, connected with the node u or v while not including edge e :

$$N(e) = \{l = (v, i) \in E \mid i \in N(v)\} \cup \{l = (u, j) \in E \mid j \in N(u)\} - \{e\}, \quad (2)$$

wherein $N(v)$ and $N(u)$ denote the neighbor node sets of node v and node u .

Definition 3. Given threshold ε , $N_\varepsilon(e)$ is the ε neighborhood of e and it means the edge set in which the similarity between e and any edge of $N(e)$ is greater than or equal to ε ; namely,

$$N_\varepsilon(e) = \{l \in N(e) \mid s(l, e) \geq \varepsilon\}. \quad (3)$$

Definition 4. Given parameters ε and μ , e will be called core edge only if the number of ε neighborhood of e is greater than or equal to μ , so

$$e.Core = \text{true} \iff |N_\varepsilon(e)| \geq \mu. \quad (4)$$

Definition 5. Edge e will be known as a boundary when it does not meet the conditions of core edge but can reach a core edge directly (existing in the ε neighborhood of a core edge); thus we can get

$$\begin{aligned} e.Border &= \text{true} \iff \\ e.Core &= \text{false} \wedge \exists l (l \in N(e)) : \\ l.Core &= \text{true} \wedge e \in N_\varepsilon(l). \end{aligned} \quad (5)$$

Definition 6. Edge e is called a noise edge if it does not meet the conditions of a core edge, or does not have any connected edge, or is not directly accessible to all connected edges:

$$e.Noise = \text{true} \iff N(e) = \emptyset \vee \forall l (l \in N(e)) : \quad (6)$$

$$\text{Dirreach}(e, l) = \text{false}.$$

Definition 7. Given parameters ε and μ , when edge e is a core edge and l belongs to ε neighborhood of e , it can be said that l is directly density reachable to e ; namely,

$$\text{DirReach}(e, l) \iff e.Core = \text{true} \wedge l \in N_\varepsilon(e). \quad (7)$$

Direct density reach is a kind of asymmetric relationship and only when the two edges are both core edges, it meets the symmetrical relationship.

Definition 8. Given parameters ε and μ , if there is an edge chain $e_i, e_{i+1}, \dots, e_j$, $e = e_i$, $l = e_j$, and any e_{i+1} is directly density reachable to e_i ($i < j$), l is density reachable to e :

$$\begin{aligned} \text{Reach}(e, l) &\iff \exists e_i, e_{i+1}, \dots, e_j \in E : \\ e &= e_i \wedge l \\ &= e_j \wedge \forall i \ (i < j) : \text{DirReach}(e_i, e_{i+1}). \end{aligned} \quad (8)$$

Density reach has transitivity and its essence is a transitive closure of direct density reach. However, if one of the two edges is not core edge, density reach has no asymmetric relationship.

Definition 9. Given parameters ε and μ , edge e density connects to edge l only when edges e and l are density reachable to t , which means

$$\begin{aligned} \text{Connect}(e, l) &\iff \exists t \in E : \\ \text{Reach}(t, e) &\wedge \text{Reach}(t, l). \end{aligned} \quad (9)$$

Density connection has symmetry. If edges e and l are density connected, so are edges l and e .

Definition 10. Given parameters ε and μ , link community LC refers to a nonempty edge subset which satisfies the following two conditions:

- (1) connection: any two edges are connected in LC; that is, $\forall e, l \in \text{LC} : \text{Connect}(e, l)$;
- (2) maximum: $\forall e, l \in E : e \in \text{LC} \wedge \text{Reach}(e, l) \Rightarrow l \in \text{LC}$.

According to Definition 10, the first step of DBLINK algorithm is to find all the link communities in a network meeting connection and maximum. If one edge does not belong to any link community, the edge is called an isolated edge. Any two edges within the link community are density connected, and meet the symmetrical relationship; therefore, given parameters ε and μ , DBLINK can start from any edge when detecting link communities and the result is uniquely determined.

Definition 11. Node community C means all nodes connected by edges in the corresponding link community LC:

$$\begin{aligned} \forall e = (u, v) \in \text{LC} &\implies \\ u &\in C \wedge v \in C. \end{aligned} \quad (10)$$

Definition 12. If there are two edges connecting the same node v , respectively, belonging to different link communities, then node v is called overlapping node, so

$$\begin{aligned} \text{Overlap}(v) \\ \iff \exists u_1, u_2, \text{LC}_1, \text{LC}_2 \ (u_1 \neq u_2 \wedge \text{LC}_1 \neq \text{LC}_2) : \\ e = (v, u_1) \in \text{LC}_1 \wedge e = (v, u_2) \in \text{LC}_2. \end{aligned} \quad (11)$$

The second step of DBLINK algorithm is to convert the link community set LCS to the corresponding node community set CS based on Definitions 11 and 12.

FIGURE 1: Demonstration of link communities.

As shown in Figure 1, we can get two link communities given parameters $\varepsilon = 0.5$ and $\mu = 4$, and also edge (1, 7) is an isolated edge which do not belong to any link community. As a result, the final node communities are $\{0, 1, 2, 3, 4, 5\}$ and $\{5, 6, 7, 8, 9\}$, where node 5 is an overlapping node. We have done a lot of experiments in the literature [6] and proved that the DBLINK algorithm not only gives consideration to the time efficiency, but also finds the overlapping community structure with high quality in the network.

3. Dynamic Community Detection iDBLINK

An incremental density-based link clustering algorithm iDBLINK for community detection in dynamic networks has been proposed in order to effectively track and analyze the community structure of temporal dynamic network. This algorithm is an extended version of DBLINK which can effectively update the current community structure according to the change of the community structure and network topology a moment before.

3.1. Problem Description. It can be assumed that a complex network is described by an undirected and unweighted graph $G = (V, E)$, where V represents node set and E is edge set. $C = \{C_1, C_2, \dots, C_k\}$ means the community structure of the networks and $\forall C_i \in C$ is a nonempty subset of V .

$G_0 = (V_0, E_0)$ represents the original network, $G_t = (V_t, E_t)$ denotes the network snapshot at time t , ΔV_t and ΔE_t , respectively, refer to the set of nodes and edges joined or deleted at time t , and $\Delta G_t = (\Delta V_t, \Delta E_t)$ means the change of the whole network. The network snapshot at time $t + 1$ is $G_{t+1} = G_t \cup \Delta G_t$. A dynamic network model $\mathcal{G}$ is a sequence of the network snapshots at different time; thus $\mathcal{G} = \{G_0, G_1, G_2, \dots\}$.

Given a set of dynamic network model $\mathcal{G} = \{G_0, G_1, G_2, \dots, G_s\}$, G_0 represents the original network and $G_1, G_2, \dots, G_s$ is the network snapshots according to $\Delta G_1, \Delta G_2, \dots, \Delta G_s$. Incremental community detection intends to design an adaptive algorithm to effectively update the current network community structure according to the previous moment network snapshots and the changes and also track the evolution of the network, just as shown in Figure 2.

The mission in the first phase of incremental community detection is to cluster the original network G_0 , so as to find the community structure and lay the foundation for community updating of the network snapshots. We select DBLINK as the

FIGURE 2: Incremental community detection model.

FIGURE 3: Framework of iDBLINK.

community detection algorithm during the initial stage. This algorithm has advantages of high time efficiency, as well as the ability to find high quality overlapping community structure. We will prove it in the experiments.

3.2. Algorithm Framework. Figure 3 shows the iDBLINK algorithm framework and it can be seen that iDBLINK focuses on the change of link communities. This has two advantages: first, it reduces the factors needed to be considered because link communities are nonoverlapping; second, it naturally updates changes of overlapping node community according to the variation of nonoverlapping link community.

3.3. Network Topology Changes. iDBLINK is an extension of DBLINK in essence and the deciding factor of link community structure is the similarity between edges; therefore, we need to further translate the changes of network topology into the changes of similarity. For instance, the enhancement of similarity relation between the two edges may make one edge enter into the ε neighborhood of another edge and make it become a core edge, resulting in the growth of link communities or the consolidation of multiple link communities.

Assuming that edge $e = (u, v)$ is a change of the network topology, namely, edge e needs to be added or deleted, then the changes of the similarity between the network edges caused by edge e are divided into the following several situations:

- (1) The similarity between edge e itself and its neighbors will change. When edge e is the new added edge, the similarity between itself and its neighbors starts from

scratch; when edge e is to be deleted, the similarity between them is just the opposite.

- (2) If edge $e_1 = (u, v_1)$ is connected to node u and edge $e_2 = (v_1, v_2)$ is linked to edge e_1 ($v_2 \neq u$), the similarity between e_1 and e_2 will change. Node v added to $\Gamma(v_1)$ is bound to cause the change of $|\Gamma(u) \cup \Gamma(v_2)|$ and $|\Gamma(u) \cap \Gamma(v_2)|$ also may change.
- (3) If edge $e_1 = (u, v_1)$ and edge $e_2 = (u, v_2)$ are the two edges, respectively, connected to node u , similarity between e_1 and e_2 will not change. This is because $\Gamma(v_1)$ and $\Gamma(v_2)$ which are determinants of similarity between e_1 and e_2 have no changes, and the value of $S(e_1, e_2) = |\Gamma(v_1) \cap \Gamma(v_2)| / |\Gamma(v_1) \cup \Gamma(v_2)|$ keeps invariant. The two edges connected to node v appear at the same situation.
- (4) If two edges do not connect to each of node u or v , namely, $e_1 = (w, v_1)$, $e_2 = (w, v_2)$, and $u \notin \{w, v_1, v_2\} \wedge v \notin \{w, v_1, v_2\}$, the similarity between edge e_1 and e_2 has no changes also because of the unchanged $\Gamma(v_1)$ and $\Gamma(v_2)$.

In conclusion, if edge e is a new adding edge or is already removed, edge e needs to create or cancel the similarity with its neighbors. Assume that edge l is the edge which is connected to node u or v . The similarity between edge l and the edges associated with l will also change. In the actual calculation, some unnecessary repeated computation can be avoided considering the symmetry of similarity.

We first intuitively analyze the effects of link community structure by the change of similarity between two edges in the following network. Assume that the similarity of two edges e and l is $s^{(t)}(e, l) < \varepsilon$ at t moment and their connection strengthens at $t + 1$ moment, just as $s^{(t)}(e, l) < s^{(t+1)}(e, l)$. If $s^{(t+1)}(e, l) \geq \varepsilon$, one or two of edges e and l may be the core edge and ε neighborhood will also be changed: (1) increasing an existing community; (2) forming a new community; (3) building bridges between the two existing nonoverlapping communities and merging them. On the contrary, if the similarity between the two edges becomes smaller as $s^{(t)}(e, l) > s^{(t+1)}(e, l)$ and $s^{(t+1)}(e, l) < \varepsilon$, ε neighborhood of the two edges will be contracted. The two edges may not be density connected which will cause the division or disappearance of communities.

We will divide the changes of link communities into positive changes and negative changes. Positive changes mean the growth, merge, or creation of link communities; negative changes are the division, recession, or disappearance of link communities.

3.4. Update Positive Changes. We assume that during the two adjacent moments t and $t + 1$, the similarity of edge e and edge l is $s^{(t)}(e, l) < \varepsilon$, and $s^{(t+1)}(e, l) \geq \varepsilon$; namely, the similarity between edges e and l gets strengthened. This will probably cause the several circumstances in Figure 4.

- (1) **New Core Edge.** At least one of edge e and edge l is core edge. iDBLINK constructs a new neighborhood relation using the new core edge e (or l) and its neighbors (including

FIGURE 4: Update link communities for positive changes.

l) and updates accordingly by examining whether edge e or l has been assigned a community label.

(2) *New Boundary Edge*. An original noise edge e becomes a boundary edge, namely, direct density reach to a core edge l ; iDBLINK will assign edge e to the community of edge l .

(3) *New Noise Edge*. The addition of a new edge does not meet the above two cases; then the link community structure remains unchanged.

From Figure 4, we can see that only when producing new core edge or new boundary edge, it updates the corresponding link communities. The following is mainly about the creation, growth, and merging of communities.

(1) *Creation of New Link Communities*. When a noise edge (not belonging to any existing link communities) turns into the core status, a new link community will come into being and iDBLINK will assign a new community label to this core edge and its corresponding ϵ neighborhood.

(2) *Growth of Link Communities*. If edge e contains a new ϵ neighborhood at $t + 1$ moment and has been allocated a link community label at t moment, then ϵ neighborhood of edge e will be absorbed by the link community of this label, so that link communities will get growing.

(3) *Merging of Link Communities*. Suppose edge e becomes a core edge at $t + 1$ moment and the edges in the field of ϵ neighborhood belong to different link communities. iDBLINK incorporates these different link communities and assigns them a new community label. It is worth mentioning that the generation of a new core edge can only cause the merge of μ link communities at most.

3.5. Update Negative Changes. We assume that during the two adjacent moments t and $t + 1$, the similarity of edge e and edge l is $s^{(t)}(e, l) \geq \epsilon$, and $s^{(t+1)}(e, l) < \epsilon$; namely, the similarity between edges e and l gets weakened. This will probably cause the several circumstances in Figure 5.

(1) *Losing the Core Edge*. Edge e is a core edge at t moment, while at $t + 1$ moment it loses the core attribute because of its lower similarity with edge l ; namely, $|N_\epsilon(e)| < \mu$.

(2) *Losing Connection with the Core Edge*. Edge e is the boundary of edge l at t moment, but it loses relationship of density direct reach with l at $t + 1$ moment.

(3) *Losing Connection between Two Core Edges*. Edge e and edge l are all core edges at t and $t + 1$ moment, and also they are density connecting at t moment, while losing it at $t + 1$ moment.

(4) *Losing Connecting Relationship*. Two connected edges e and l are not core edges at t moment, and they lose connection at $t + 1$ moment. At this time link communities have no change.

For the above four conditions except for the last one, iDBLINK will calculate the ϵ neighborhood of affected core edges again, and detect the distribution of link communities within ϵ neighborhood. There are three types of the corresponding operations as follows.

(1) *Deletion of Link Communities*. Assuming that edge e is an edge losing core attributes, there is no other core edges existing in ϵ neighborhood; then the original link community where edge e is in will be deleted. The original members in the link community set the status as noise or boundary according to the specific situation.

(2) *Contraction of Link Communities*. Suppose edge u is in the ϵ neighborhood of core edge l at t moment; it loses connection with edge l at $t + 1$ moment, but it has not changed the core attribute of edge l . If so, edge e will be deleted from the link community of edge l , and link community size becomes smaller, just as link community contraction.

(3) *Division of Link Communities*. Assume that edge e and edge l are all core edges at t and $t + 1$ moment, and density

FIGURE 5: Update link communities for negative changes.

FIGURE 6: Demonstration of overlapping node community formation.

connected at t moment. However, they lose this relationship at $t + 1$ moment, so link communities will be split at this time.

3.6. Node Community Update. Although iDBLINK only cares about updating link communities in dynamic network, and link community structure is nonoverlapping, the corresponding overlapping node communities can be naturally reflected according to link communities. All nodes connected by edges within the same link communities belong to the same node communities, while intersection nodes that belong to different link communities are overlapping nodes. As shown in Figure 6, (a) embodies the link communities and node communities in the network at $t - 1$ moment. At moment t , an edge is added which makes the formation of new link communities, and naturally four yellow nodes become overlapping nodes connecting two communities.

4. Experimental Results

This section is to verify the performance of the iDBLINK algorithm. We will, respectively, compare iDBLINK with static community detection algorithms DBLINK, FOCS, and DENGGRAPH and dynamic community detection algorithms AFOCS and DENGGRAPH-IO. The experimental environment was as follows: processor Inter(R) Core(TM)2 Duo 2.8 GHz PC, internal storage 2 G, operating system Windows 7, and programming language C#.Net.

4.1. Experiment Settings. We plan to verify the performance of iDBLINK algorithm with four groups of LFR simulation networks and two real networks. The simulation network configurations of each group are shown in Table 1.

N is the number of nodes. k is the average degree of nodes in the network and $\max k$ represents the maximum degree. $\min c$ is the number of nodes in the minimum community

FIGURE 7: Results on S1.

and $\max c$ is the number of nodes in the maximum community. Mixing parameter μ represents the probability of nodes connected with external community. The greater the value of μ is, the more difficult the community detection will become. The community structure of LFR benchmark network is already known, so we can evaluate the quality of the community finding by different algorithms.

We selected Amazon and DBLP [23] as real networks, which are large complex networks with true value of community structure.

DBLP is a scientist cooperation network that widely applied. Scientists are as nodes in the network and if two scientists at least cooperate to publish one article, then they establish an edge between them. The network contains 317080 nodes and 104986 edges.

Amazon is goods connecting network, where goods are regarded as the nodes and if two products are frequently bought at the same time, they establish a link between the two nodes. The network has 334863 nodes and 925872 edges.

Both LFR benchmark networks and real networks and their dynamic networks are generated as follows: for each network, we first select 90% of the edges as the original network G_0 , and, in addition to the original network, we generate ten networks $G_1, \dots, G_{10}$ to make up dynamic networks. Then every network removes 5% of the entire network edges and adds the same amount of new edges, which means dynamic networks always keep the number of edges invariant.

4.2. Evaluation Standards. For the LFR benchmark networks, we adopt the normalized mutual information (NMI) given in [24] as the evaluation standard to compare all kinds of algorithms, since the true value of community structure is known.

We suppose that the real community set in the network is C ; community affiliation of node i can be represented as

binary vector x_i whose length is $|C|$. The fact that the value of $(x_i)_k$ is one or zero indicates whether node i belongs to the k th community. The k th element can be considered as random variable X_k , and its probability distribution is $P(X_k = 1) = n_k/n$, $P(X_k = 0) = 1 - n_k/n$ where n_k means the number of nodes in the k th community and n represents the number of all nodes in the network. Also, in the community set C' found by the algorithm, Y_l is used to represent the probability distribution of nodes in the l th community. $H(X_k | Y_l) = H(X_k, Y_l) - H(Y_l)$ is defined as the conditional entropy of X_k in Y_l , and, according to $H(X_k | Y_l)$, the conditional entropy $H(X_k | Y)$ of X_k in Y (collection composed by Y_l) is

$$H(X_k | Y) = \min_{l \in \{1, 2, \dots, |C'|\}} H(X_k | Y_l). \quad (12)$$

Standardization of conditional entropy of X (collection composed by X_k) in Y is

$$H(X | Y) = \frac{1}{|C|} \sum_k \frac{H(X_k | Y)}{H(X_k)}. \quad (13)$$

Similarly, we can calculate the standardization conditional entropy $H(Y | X)$ of Y in X . According to formula (14), we can calculate the normalized mutual information $\text{NMI}(X | Y)$ of two community sets:

$$\text{NMI}(X | Y) = 1 - \frac{[H(X | Y) + H(Y | X)]}{2}. \quad (14)$$

4.3. Experimental Results of Simulative Network. Figures 7–10 show the comparison of six algorithms. The abscissa represents the network at each moment, zero coordinates refer to the original network, and ordinate, respectively, shows the community detection quality NMI (a) and the algorithm running time (b).

We first compare the two static algorithms DBLINK and FOCS on the network. At time $t = 0$, community detection

TABLE 1: LFR benchmark network information.

Number	N	k	$\max k$	$\min c$	$\max c$	μ	on	om
S1	5000	10	50	10	50	0.1	100	2
S2	5000	10	50	10	50	0.3	100	2
S3	5000	10	50	10	50	0.1	500	2
S4	5000	10	50	10	50	0.3	500	2

FIGURE 8: Results on S2.

findings of four simulative networks show that the quality of the community which DBLINK finds is better than FOCS; especially on the two lower overlap networks, advantages are more obvious. However, the time efficiency of DBLINK algorithm is slightly lower than FOCS.

Next, we focus on the comparison of each algorithm for dynamic network in time efficiency and quality of community detection. FOCS and DBLINK, respectively, conduct community detection at every moment on the whole network, and iDBLINK, AFOCS are all incremental community detection algorithms.

- (1) In terms of the algorithm efficiency, DBLINK and FOCS all run algorithm on the entire network, so its program running time is substantially consistent with the running time on the network at time $t = 0$. Nevertheless, since iDBLINK and AFOCS quickly update local community structure according to changes of the network topology (10% of edges added or deleted), their running time is significantly less than that of the corresponding static network community detection algorithms. Moreover, the time efficiency of DBLINK algorithm is slightly lower than FOCS. On the contrary, we believe iDBLINK relative to DBLINK in the increase of time efficiency is greater than AFOCS to FOCS.
- (2) In terms of the community detection quality, we can see that the community detection results of AFOCS

and FOCS do not always coincide; even the NMI values of AFOCS appear to be greater than FOCS, while iDBLINK is almost the same as DBLINK at NMI values of community structure found at every moment on the network. It shows that iDBLINK is indeed the effective incremental community detection version of DBLINK. In most cases, especially on the two lower overlap networks, quality of community structure detected by iDBLINK algorithm is superior to that of AFOCS.

In conclusion, the iDBLINK algorithm proposed in this paper not only keeps the high quality community structure detection performance of DBLINK algorithm, but also has a very good time efficiency on the dynamic networks.

4.4. Experimental Results of Real Networks. We have further validated the performance of iDBLINK on the two real networks DBLP and Amazon. These two networks are all very large complex networks. AFOCS and FOCS fail at first step due to the insufficient memory, so, as shown in Figures 11 and 12, we can only get the results of DENGGRAPH-IO and iDBLINK. It is obvious that the result of iDBLINK is better comparing with DENGGRAPH-IO and is very close to the community detection quality of DBLINK algorithms. However, its time efficiency is far superior to DBLINK algorithm, which explains that iDBLINK is indeed the effective

FIGURE 9: Results on S3.

FIGURE 10: Results on S4.

incremental version of DBLINK and can be very effectively used in the community detection on dynamic networks.

5. Conclusion

This paper proposes an incremental density-based link clustering algorithm for community detection in dynamic networks iDBLINK. This algorithm can update the corresponding community structure quickly and efficiently according to changes of network topology. It decomposes the change of the network into four cases: adding and deleting nodes and adding and deleting edges. Furthermore, this algorithm adjusts the link density in each network for

each case, and, based on the connectedness of density-based algorithm, it updates the current community structure quickly and efficiently according to the community structure at previous moment. iDBLINK just updates the current community structure in view of the network topology changes, so iDBLINK can be effectively applied to community detection in the dynamic network. In addition, iDBLINK has kept the advantages of DBLINK algorithm, that is, to find overlapping community structure with high quality. Experimental results show that iDBLINK not only has a great time efficiency, but also maintains a high quality community detection performance while the network topology is changing.

FIGURE 11: Results on DBLP.

FIGURE 12: Results on Amazon.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (Grant no. 2012AA0622022 and Grant no. 2012AA011004), the Doctoral Fund of Ministry of Education of China (Grant no. 20100095110003 and Grant no. 20110095110010), the Fundamental Research Funds for the Central Universities (under Grant no. 2013XK10), the National Natural Science Foundation of China (Grant no. 61402482), China Postdoctoral Science Foundation (no. 2015T80555), and Jiangsu Planned Projects for Postdoctoral Research Funds (no. 1501012A).

References

- [1] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Statistical Analysis and Data Mining*, vol. 4, no. 5, pp. 512–546, 2011.
- [2] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [3] W. Lin, F. Lu, and Z. Ding, "Parallel computing hierarchical community approach based on weighted-graph," *Journal of Software*, vol. 23, no. 6, pp. 1517–1530, 2012.
- [4] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [5] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

- [6] M. Zhu, F. Meng, and Y. Zhou, "Density based link clustering algorithm for overlapping community detection," *Journal of Computer Research and Development*, vol. 50, no. 12, pp. 2520–2530, 2013.
- [7] T. S. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, Article ID 016105, pp. 145–148, 2009.
- [8] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi, "Multidimensional networks: foundations of structural analysis," *World Wide Web*, vol. 16, no. 5, pp. 567–593, 2013.
- [9] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, Article ID 033015, 2009.
- [10] N. P. Nguyen, T. N. Dinh, D. T. Nguyen et al., "Overlapping community structures and their detection on social networks," in *Proceedings of the IEEE 3rd International Conference on Privacy, Security, Risk and Trust (PASSAT '11), and IEEE 3rd International Conference on Social Computing (SocialCom '11)*, pp. 35–40, IEEE, Boston, Mass, USA, October 2011.
- [11] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, Article ID e18961, 2011.
- [12] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.
- [13] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 2004.
- [14] Z. Ye, S. Hu, and J. Yu, "Adaptive clustering algorithm for community detection in complex networks," *Physical Review E*, vol. 78, no. 4, Article ID 046115, 2008.
- [15] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, Article ID 036106, 2007.
- [16] J. Xie and B. K. Szymanski, "LabelRank: a stabilized label propagation algorithm for community detection in networks," in *Proceedings of the IEEE 2nd Network Science Workshop (NSW '13)*, pp. 138–143, IEEE, West Point, NY, USA, May 2013.
- [17] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [18] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proceedings of the 11th IEEE International Conference on Data Mining Workshops (ICDMW '11)*, pp. 344–349, IEEE, Vancouver, Canada, December 2011.
- [19] J. Xie, M. Chen, and B. K. Szymanski, "LabelRankT: incremental community detection in dynamic networks via label propagation," in *Proceedings of the Workshop on Dynamic Networks Management and Mining (DyNetMM '13)*, pp. 25–32, ACM, New York, NY, USA, June 2013.
- [20] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai, "Adaptive algorithms for detecting community structure in dynamic social networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '11)*, pp. 2282–2290, Shanghai, China, April 2011.
- [21] T. Falkowski, A. Barth, and M. Spiliopoulou, "Dengraph: a density-based community detection algorithm," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 112–115, IEEE, November 2007.
- [22] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai, "Overlapping communities in dynamic networks: their detection and mobile applications," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, pp. 85–95, ACM, September 2011.
- [23] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (MDS '12)*, article 3, Beijing, China, August 2012.
- [24] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 551–556, ACM, Blacksburg, Va, USA, August 2004.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>