A community detection algorithm based on node degree difference and node similarity

ABSTRACT

Detecting communities in complex networks is important for discovering the relationships between networks' structure and networks' functions. Though there are many algorithms to detect community structures from complex networks, how to get a higher quality community structure is still a challenging problem. This paper proposes an algorithm called DDSCDA, which is based on the concepts of the node degree difference and the node similarity. In the algorithm, we iteratively extract the node from the network with larger degree and certified the node as a kernel node, then take the kernel node as the founder or initiator of a community to attract its neighbors to join in that community; by doing so, we obtain a partition corresponding to a coarse-grained community structure of the network, in which, every node in each community has a label that the kernel node has. Finally taken the coarse-grained community as a starting point, we use the strategy of LPA to propagate labels through the network further. At the end of the algorithm, all the nodes with the same labels are grouped as a community logically, and all the communities constitute the final community structure. We have tested our algorithm on several real networks, and compared the performance with classical community detection algorithms such as LPA, LPAm, Fast$Q$, etc., the experimental results have manifested that our proposal is a feasible algorithm, can extract higher quality communities from the network, and outperforms the previous algorithms significantly.

Categories and Subject Descriptors

H.2.8 [**Database Application**]: Data Mining

General Terms

Theory, Algorithm

Keywords

Community detection, Community structure, Node degree difference

1. INTRODUCTION

A complex network consists of many nodes and edges, in which each edge represents the relationship of the interaction, the tie or connection, etc. between the two nodes associated with the edge. Many complex networks have presented a common feature — community structure, which discovered the network's topological property that the connections between nodes in the same community are much denser than those between nodes across the different communities. To detect the community structure from a network is of great significance, because the community structure of a network can help us to discover the relationships between the structure component and functions of the network. So how to identify the communities in complex networks has attracted more and more researchers' attention and interests.

The problem of community detection can be traced back to the problem of graph partitioning in graph theory proposed by Kernighan and Lin in 1970 [11], since then, along with the development of the complex network theory, many researchers have proposed many methods and algorithms to detect the community structure of the networks, such as spectral algorithm [7, 18], Random walking method [17], sort method [16] and heuristic graph partition method [11] based on the graph theory, agglomerative methods based on the structures' similarity [6], Potts model method [8, 21], etc. Among them, the most famous one is the GN algorithm proposed by Girvan and Newman [10, 15]; this algorithm calculates the betweenness for all the edges in the network, and removes the edge with the highest betweenness in each iteration; this process continues iteratively until all the edges are removed from the network. It is a divisive algorithm, and utilizes the concept of modularity Q [10, 1, 2, 5] to measure the strength of the community structure. The disadvantage of this algorithm is its high computation complexity, so that this algorithm cannot be used to process very large networks. Radicchi et al. proposed another algorithm, into which the edge-coefficient was added, to reduce the computation complexity [19]; this is also a divisive algorithm; its computation complexity is still high. To increase the computing efficiency, Newman proposed a greedy algorithm named as Fast$Q$ [14] to detect the community structure from networks; it approached the task from a perspective of modularity optimization, in which, starting with each node being the sole member of a community of one, the algorithm repeatedly join together a pair of communities whose merge should lead to the greatest increase or smallest decrease of modularity Q, until all the nodes are merged into the same giant community. Clauset et al. proposed the CNM algorithm [4], which can be regarded as an improvement version of the Fast$Q$ algorithm, to some extent. Its idea is identical with the Fast$Q$ algorithm, but the CNM algorithm exploited a sophisticated strategy and data structures to implement the algorithm, and the new implementation led to the result that it runs far more quickly than Fast$Q$. Latter, Raghavan [20] proposed a method called label propagation algorithm (LPA) to identify community structure of a network, by updating every node's label to the one most neighbors of that node have iteratively, until all the nodes and most of their neighbors have the same label. So that, the labels are propagated through the whole network, and at the end of the process, all the nodes have the same label formed a community. LPA has near linear time complexity, but it is not a deterministic algorithm, that is, running the algorithm upon a given network many times, the result community structures extracted from the network may not be identical. In spite of this, just because of its computation efficiency, many variations and improvements of LPA have been brought out, and the LPAm [1] is a representative algorithm among them.

In this paper, we propose a new concept — node degree difference; and based on this concept and the node similarity measure, we propose a new community detection algorithm called DDSCDA, which is abbreviation for node Degree Difference and Similarity based Community Detection Algorithm. Just as its name implies, this algorithm makes utilization of the concept of node degree difference and the node similarity to identify community structure from the net-work. Compared with the GN algorithm, the computation complexity of the DDSCDA is relatively lower than that of the GN algorithm, and compared with the LPA based algorithm, the DDSCDA is a deterministic algorithm.

The remainder of this paper is organized as follows: our proposed algorithm is elaborated in section 2; section 3 is the detailed description of the experiments and results analysis; and finally we make a summarization and conclusion in section 4.

2. PORPOSED ALGORITHM

In this section, we put forward our proposal — the DDSCDA algorithm, which is designed to identify high quality community structures from the networks quickly.

2.1 Intuitions and observations

The algorithm is based on some intuitions and observations following:

- LPA is a non deterministic algorithm, this is because the algorithm breaks ties uniformly randomly; however, in the early iterations of the algorithm, the possibilities of ties are very high, hence, the algorithm may produce different community structures in different runs. So, intuitively, if we can manage to get a coarse-grained partition of the network using deterministic methods, and take that partition as the initial state to propagate the labels in the network, instead of that of LPA in which every node is the sole member of each community, the possibilities of ties can be reduced substantially, and then the algorithm will tend to yield the results with much more certainty.

- In each community of the true community structures, there are always some nodes with higher degree than the other nodes. In social networks, those nodes might correspond to some people with larger influential, and the corresponding communities can be regarded as cliques formed around those people. From this perspective, the node with largest degree in each community is the kernel of that community. When considering the forming process of every community in the coarse-grained community structures, we can extract the kernel from the network firstly, and then take it as the founder or representative of the community to attract its neighbors to join in that community.

- To every neighbor of the kernel, if the difference between the kernel's degree and the neighbor's degree is large enough, the neighbor node's influential in the network is overshadowed by the kernel; in fact, that neighbor node and even some of its neighbors should belong in the same community with the kernel, i.e., the difference between kernel's degree and its neighbor's degree is of great importance to the establishment of the coarse-grained community structure. So, we formulate it as a new concept — node degree difference in this paper. On the other hand, if the difference is not so large, whether the neighbor node belongs in the kernel's community is dependent on the similarity between the two nodes, so we make utilization of the combination of the concepts of node degree difference and node similarity to construct the coarse-grained community structure.

Based on the intuitions and observations mentioned above, we bring forth our algorithm.

2.2 Definitions

To facilitate the description of the algorithm, some definitions are given below; especially, our proposed new concept of node degree difference is also among them.

Definition 1. A network is a graph $G(V, E, W)$, where, V and E are the node set and the edge set respectively, and $|V| = n$, $|E| = m$; W is the adjacency matrix of the network, i.e., W_{uv} is the weight of the edge (u, v); for an unweighted network, $W_{uv} = 1$ for all the $u \in V, v \in V$, and $(u, v) \in E$; and $W_{uv} = 0$, otherwise [3]. Based on this, we extend the concept of node degree to be sum of the weights of the edges incident upon it, that is

$$d_u = \sum_{v \in V} W_{uv} \tag{1}$$

Definition 2. For a given node $u \in V$, the neighbor set of node u is defined as

$$N_u = \{v | v \in V, (u, v) \in E\} \tag{2}$$

which is a set contains all the neighbors of the node u.

Definition 3. In our proposed algorithm, we need to make utilization of the similarity between two nodes, any form of similarity measure can take effect. In this paper, we define the similarity between node u and node v as the ratio of the number of their shared neighbors to the total number of both of their neighbors, that is

$$sim(u,v) = \frac{|N_u \cap N_v|}{|N_u| + |N_v|} \tag{3}$$

Definition 4. For two nodes u and v connected by an edge, the node degree difference between them is define as

$$NDD(u,v) = \frac{|d_u - d_v|}{max(d_u, d_v)} \tag{4}$$

2.3 Proposed algorithm

Our proposed DDSCDA is a two-stage algorithm, the first stage is to construct the coarse-grained community structure, the second stage is to fine tune the community structure to get the final result.

In the first stage, the initial setting of the algorithm is that all the nodes are marked as "unclassified", and every node has a distinct label. Then, we extract the kernel node and use the kernel to attract some of its neighbors to join in the community iteratively. A kernel is a node with largest degree among the nodes in its community, but whether a node can be certified to be a true kernel or not does not depend on its own degree only. We select the node with the largest degree from the "unclassified" node set as a kernel candidate, if majority of the node's neighbors are not classified yet, the candidate is qualified to be a true kernel; on the contrary, if most neighbors of the node have been marked as "classified", and among the classified neighbors, if the number of nodes with the most same label is larger than those of unclassified neighbors, according to the majority voting strategy, that node should be a member of the community in which its neighbors with most same label belong; hence, the label of that node is updated to be the one most of its neighbors have and that node is marked as "classified". However, if the number of classified neighbors with the most same label is less than those of unclassified neighbors, that node can be regarded as a kernel, temporarily.

Determination of a node being a kernel means a new community is established, and the kernel is the initiator of that new community. Next, we use the kernel as the representative of the community to attract some of its neighbors to join in that community. For every neighbor of the kernel, if the degree difference between the kernel and that neighbor node is larger than a given threshold , that neighbor is dominated by the kernel, and should be a member of the kernel's community; to reflect this relationship, its label is updated to be the kernel's label, and it is marked as "classified"; otherwise, the relationship between the neighbor node and the kernel is dependent on the similarity between them, i.e., if the similarity between them is larger than a given threshold , then the label of the neighbor node is updated to the kernel's label and the neighbor node is marked as "classified", means that the neighbor node is attracted by the kernel to join in the kernel's community; else leave that neighbor node untouched.

This process is performed iteratively, until all the nodes in the network are marked as "classified". By doing so, we obtain a partition of the network corresponding to the coarse-grained community structure at the end of the first stage, in which each community consists of all the nodes with the same label.

Pseudo-code outlining this procedure is shown in the algorithm 1, in which, V_c and V_u represent the "classified" node set and "unclassified" node set, respectively; Line 3 selects the node with the largest degree from "unclassified" node set as a kernel candidate; line 4 – 10 make judgement whether the kernel candidate can be qualified to be a true kernel; the satisfaction of condition in line 6 means the kernel candidate cannot be accepted as a true kernel, then the operations in line 7 – 8 update its label to be the one of most of its neighbors have and mark it as "classified", i.e., the kernel candidate is inserted into an existed community; the operation in line 12 assigns a new label to the kernel certified just now, which means a new community is established; line 13 – 16 take the kernel as the initiator of the community to attract its neighbors to join in that community. At the end of the algorithm 1, all the nodes with the same label are grouped as a community implicitly, and all the communities constitute the coarse-grained community structure.

Algorithm 1: the first stage of **DDSCDA** to acquire the coarse-grained community structure

Input: $G(V, E, W)$: network; ε, σ: degree difference threshold and similarity threshold

Output: A logical partition of networks corresponding to the coarse-grained community structure

```
1  $V_c \leftarrow V; V_u \leftarrow V;$
2  while $notempty(V_u)$ do
3      $kc \leftarrow \arg\max_{v \in V}(d_v);$
4      if $|N_{kc} \cap V_u| > |N_{kc}|/2$ then
5          $k \leftarrow kc;$
6      else if $\max_{v \in N_{kc} \cap V_c}(|\{u|u \in N_{kc} \cap V_c, u.label = v.label\}|) \geq |N_{kc} \cap V_u|$ then
7          $n \leftarrow \arg\max_{v \in N_{kc} \cap V_c}(|\{u|u \in N_{kc} \cap V_c, u.label = v.label\}|);$
8          $kc.label \leftarrow n.label; V_c \leftarrow V_c \cup \{kc\}; V_u \leftarrow V_u \backslash \{kc\};$
9          goto 2;
10     else
11         $k \leftarrow kc;$
12     $k.label \leftarrow newlabel;$
13     foreach $v \in N_k$ do
14         if $(NDD(k,v) \geq \varepsilon)$ or $(NDD(k,v) < \varepsilon$ and $sim(k,v) \geq \sigma)$ then
15             $v.label \leftarrow k.label; V_c \leftarrow V_c \cup \{v\}; V_u \leftarrow V_u \backslash \{v\};$
```

The second stage takes the coarse-grained community structure as the starting point to propagate the label. Here, we borrow ideas from the LPA; in fact, our original intention is to improve the LPA by reducing the possibility of ties. So, just like how the LPA does, the nodes each update their labels to the one most of their neighbors have iteratively, until every node has a label that the maximum number of its neighbors have. Adapted with the status of the coarse-grained community structure being the starting point, the pseudo-code of the second stage is illustrated in the algorithm 2, and it is almost self-explanatory.

Algorithm 2: the second stage of **DDSCDA** to propagate the label in the network

Input: $G(V, E, W)$: network after the first stage, in which nodes with same labels are grouped into community implicitly and logically

Output: the final logical community structure, in which all the nodes with same labels form the a community

```
repeat
    foreach $node \in V$ do
        $n \leftarrow \arg\max_{v \in N_{node}}(|\{u | u \in N_{node}, u.label = v.label\}|)$;
        $node.label \leftarrow n.label$;
until every node has a label that the maximum number of its neighbors have;
```

To our proposed DDSCDA, the second stage can kill two birds with one stone, in fact. On the one hand, taken the coarse-grained community structure as the initial state to propagate the labels through the network, the final community structure can be extracted from the network more efficiently than those by LPA, which propagates the labels in the network from the state of each node being the unique member of the community, and the combination of the two stages can reduce the possibilities of ties substantially, so we get an more efficient, more deterministic community detection algorithm;

On the other hand, it can revise some mistakes introduced in the first stage; the target of the first stage is to construct the coarse-grained community structure quickly with a deterministic method to reduce the chance of ties, we do not have too much to consider the accuracy of node classification, so some nodes might be misclassified into the wrong community inevitably. There might be two kinds of mistakes may be introduced in the first stage, one is mis-determination of kernel node, the other one is mis-join of an ordinary node into a certain community attracted by that community's kernel node; Fig. 1 and Fig. 2 illustrate an instance for each of the two cases respectively, and only the nodes and edges of relevant are shown, irrelevant nodes and edges are omitted. In the Fig. 1(a), nodes with labels are those been classified already, and others are those not been classified yet. According to the rule of the algorithm 1, the node k is qualified to be a kernel node; however, in the next steps, the kernel node k can only attracts node v_5 and node $v_6$to join into its community, and the result is shown in the Fig. 1(b). From another perspective, if we are to determine whether the node k can be a kernel node starting from the setting illustrated in the Fig. 1(b), according to the rule of majority voting exploited by the algorithm 1 and LPA, the node k and even the node v_5 and the node v_6 should be members of the community whose nodes labeled with "1", that means the node should not be certified to be a kernel node. In the Fig. 2, the node v is pulled into the community in which the kernel node k belongs because of the satisfaction of the condition expressed by the line 14 in the algorithm 1, but in fact, it should belong in the community whose nodes labeled with "2".

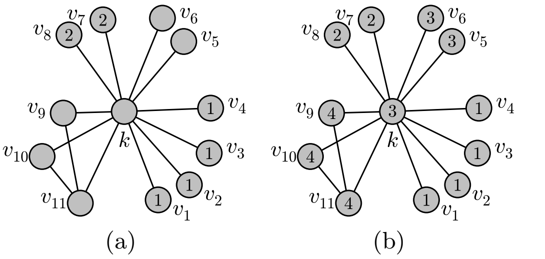


Figure 1: mis-determination of kernel node

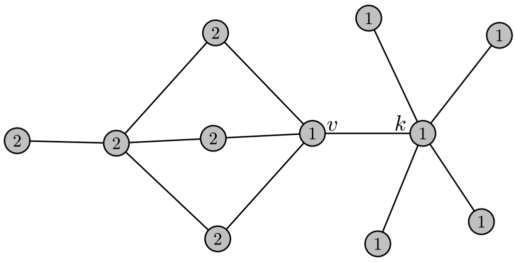


Figure 2: mis-join an ordinary node into a certain community

It is obvious that both these two kinds of mistakes can be revised by updating every node's label to the one most of its neighbors have in the second stage; and at the end of the second stage, all the nodes with the same label form a community implicitly also, and all the communities constitute the final community structure.

2.4 Time complexity

LPA is a near linear time complexity algorithm, its time consumption is $O(dn)$, where d is the average node degree of the network. Compared with it, our proposed DDSCDA also takes a near-linear time to run its completion; DDSCDA is a two stage algorithm, so the time consumption of DDSCDA is comprised of the time consumptions of the two stages naturally.

The first stage is an iterative process; in each iteration, we need to select the node with the largest degree from the unclassified node set as the kernel candidate, it takes $O(|V_u|)$ time; we assume that each kernel node absorbs nodes in its community in average in each iteration[1], so the over all time consumption of kernel candidate selection is $O(n + (n - d) + (n - 2d) + \cdots + (n - \frac{n}{d}d)) \sim O(\frac{n^2}{d})$, for n^2 is the number of edges in a complete graph with n nodes, so $O(n^2) \sim O(m)$,and $O(\frac{n^2}{d}) \sim O(\frac{m}{d}) \sim O(n)$; In each iteration, to certify the kernel candidate to be a true kernel node, we need to make judgement whether the number of the classified

[1] The number of nodes a kernel node absorbed into its community in average in each iteration is the same order of magnitude with d, so here, we assume that number is d, directly.

neighbors of the kernel candidate with most same labels are larger than the number of unclassified neighbors, it takes $O(d_{kc})$time, and the over all time consumption of kernel determination is $O(dn)$.

Although starting from the status of the coarse-grained community structure to propagate labels through the network, the second stage is a basic LPA, so it takes $O(dn)$ time complexity.

Thus, our proposed DDSCDA also takes near linear time to run its completion qualitatively. Compared with LPA, the first stage introduces some additional operations need to consume some time to execute, but taken the coarse-grained community structure as a starting point makes to reach the consensus on labels among the nodes in a community in the second stage more efficiently; in LPA, the author have found that irrespective of n, 95% of the nodes are classified correctly by the end of iteration 5; however, in our proposed DDSCDA, we have found from experiments that the algorithm can almost reach the status of convergence and at least 95% of the nodes are classified correctly by the end of iteration 2 in the second stage. Furthermore, LPA need to aggregate the results to get the final result community structure, our proposed DDSCDA need not to do that; executing the algorithm 1 and algorithm 2 sequentially only once, we can get the final result community structure.

3. EXPERIMENTS

We need to evaluate the experimental results both qualitatively and quantitatively, so the networks' true community structure must be known a priori, the scales of networks must be not too large to facilitate the interpretation and visualization of the results, and the networks must be publicly available. These limitation result in the selection of the three classical networks widely studied by many researchers: Zachary's karate club network [22], bottlenose dolphin social network [13, 12], and college football network represents the schedule of Division I games [10]; the statistical information of these three networks is listed in the table 1.

Table 1: statistical information of the networks

network	nodes	edges	communities
Karate	34	78	2
Dolphin	62	159	4
Football	15	613	13

To the parameters of the node degree difference threshold ε and the node similarity threshold σ, we have found from the experiments that when the value of ε is in the range [0.3,0.4] and the value of σ is around 0.2, we can get better result. so, in the experiments described following, we adopt the empirical value of $\varepsilon \in [0.3, 0.4]$, and $\sigma = 0.2$.

3.1 Effectiveness

To test the effectiveness of our proposal, we have run the DDSCDA on the three networks, and from each network, our algorithm can detect the known community structure reliably.

Zachary's karate club network. This is a well-known benchmark network to test community detection algorithms. It consists of 34 nodes represent the club members, and edges connect the members who were observed to interact each others both within the club and away from it. By chance, a disagreement arisen between the club's administrator and its principal instructor led to the result that the club was split into two small clubs, centered around the administrator and the instructor, respectively. Feeding this network to our proposed DDSCDA, we can get the community structure exactly matched to the club split; the community structure extracted by our algorithm is illustrated in the Fig. 3.

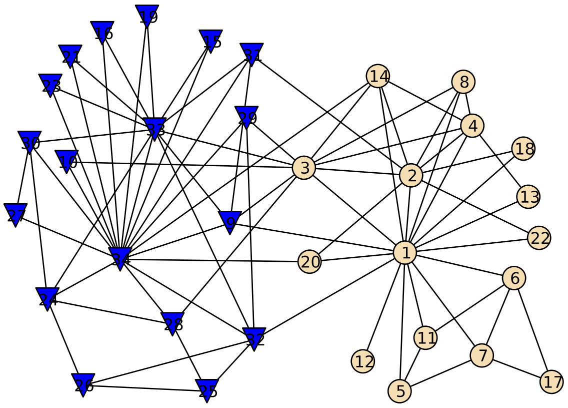


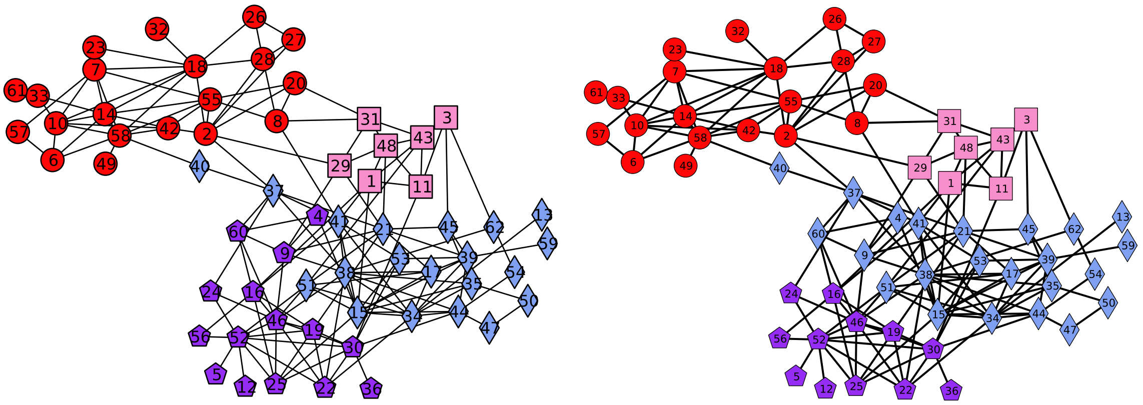
Figure 3: community structure of the karate club network

Bottlenose dolphin social network. This is also a classical network to evaluate the community detection algorithms; it contains 62 nodes represent 62 bottlenose dolphins living in Doubtful Sound, New Zealand, and 159 edges represent associations between dolphin pairs occurring more frequent statistically. The true community structure of this network is illustrated in the Fig. 4(a), this network can be split into 4 groups, represented by different node shape and shading in the figure. The community structure extracted by running our algorithm on this network is shown in the Fig. 4(b); compared with the true community structure, there are some bias in the result of our algorithm, node 60, node 9 and node 4 are classified into wrong communities.

College football network. This network is a representation of the schedule of Division I games of the United States college football for the 2000 season. In the network, there are totally 115 nodes represent 115 teams, and edges represent regular-season games between the two teams they connect; the true community structure of this network is shown in the Fig. 5(a). Running our algorithm on this network, we get the community structure illustrated in the Fig. 5(b), in which, node 59, node 98, node 60, node 64 and node 43 are misclassified.

3.2 Performance

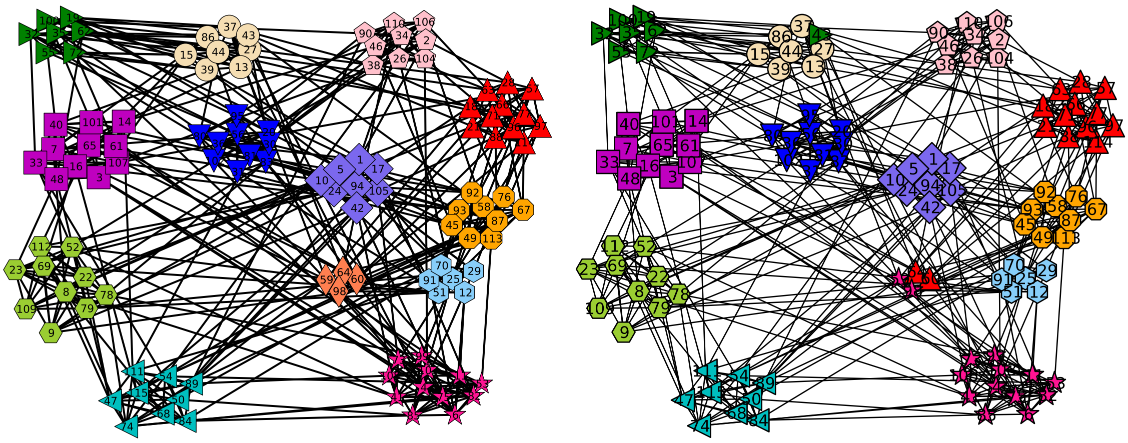
To test the performance of DDSCDA, we have also carried out the experiments on these networks by running three further algorithms, and performed the comparison on some indexes between our algorithm and the others. Those further algorithms are LPA, LPAm and Fast$Q$, and here the modularity and accuracy are employed as validate metrics.



(a) true community structure

(b) community structure detected by our algorithm

Figure 4: community structure in dophin social network



(a) true community structure

(b) community structure detected by our algorithm

Figure 5: community structure in college football network

To measure the strength of community structure, modularity is a de facto standard; however, modularity has an intrinsic scale that communities smaller than it may not be identified; that is the so-called resolution limit of modularity [9]. Besides this, the essence of community detection is equivalent to the node clustering in the network to some extent, so accuracy can also make sense to the community detection problem.

Thus, we use both modularity and accuracy as validate metrics; modularity is computed by the method in the literature [10, 15], and accuracy is defined as the ratio of the number of nodes classified correctly to the total number of nodes in the network. The comparative results is figured out in the table 2. To get the result figured out in the table, we have run LPA 100 times on each of the three networks, and aggregate the result of each run to get them; to Fast Q, we have cut the dendrogram at the point that modularity reaches its peak.

From the table 2, we can see clearly that the modularity of our algorithm is larger than those of LPAm and Fast Q, only less than that of LPA, and the accuracy of our algorithm is much higher than those of all others. This result have manifested that our proposed DDSCDA is a feasible algorithm to detect community from networks, and its performance is better than the comparative algorithms.

4. CONCLUSIONS

In this paper, we have proposed a two stage algorithm to detect the community structure from the networks. In the first stage, we construct the coarse-grained community structure by repeatedly extracting the kernel node, and taking the kernel node as an initiator of a community to attract its neighbors to join into that community utilizing the concept of node degree difference and node similarity; in the second stage, taken the coarse-grained community structure as a starting point, we borrow some ideas from LPA to propagate the labels through the network to get the final community structure efficiently.

The combination of these two stages reduces the chances of ties in LPA substantially, and as a result, we get a more deterministic, more efficient community detection algorithm, and the experimental results have manifested that our proposal outperforms the comparative algorithms significantly.

Table 2: the comparative results of the algorithms

network	LPA		LPAm		Fast$Q$		DDSCDA	
	A	Q	A	Q	A	Q	A	Q
Karate	85.29%	**0.3991**	97.06%	0.3599	97%	0.36	**100%**	0.3714
Dolphins	72.58%	0.4301	72.58%	0.3835	85.40%	0.492	**98.39%**	**0.5258**
Football	65.22%	0.5483	80.00%	0.5781	53.91%	0.5491	**95.65%**	**0.58**

5. ACKNOWLEDGMENTS

This work is partially supported by the Fundamental Research Funds for the Central University (grant id: lzujbky-2012-212).

6. REFERENCES

[1] M. J. Barber and J. W. Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2):026129, 2009.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[3] D. Chen, M. Shang, Z. Lv, and Y. Fu. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A: Statistical Mechanics and its Applications*, 389(19):4177–4187, 2010.

[4] A. Clauset. Finding local community structure in networks. *Physical review E*, 72(2):026132, 2005.

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

[6] L. Danon, A. Díaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(11):P11010, 2006.

[7] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[8] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

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

[10] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[11] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 1970.

[12] D. Lusseau and M. E. Newman. Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271(Suppl 6):S477–S481, 2004.

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

[14] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[15] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[16] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[17] P. Pons and M. Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

[18] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990.

[19] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.

[20] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[21] J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters*, 93(21):218701, 2004.

[22] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.