A link density clustering algorithm based on automatically selecting density peaks for overlapping community detection

A Link Density Clustering Algorithm based on Automatically Selecting Density Peaks For Overlapping Community Detection

Lan Huang[1,2]

Guishen Wang[1,2]

Yan Wang[1,2,a]

[1] *College of Computer Science and Technology, Jilin University,*
[2] *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education,Jilin University, Changchun, 130012, China*

[a] *wy6868@jlu.edu.cn*

Wei Pang[3]

[3] *School of Natural and Computing Sciences, University of Aberdeen, Aberdeen, AB24 3UE, UK*

Qin Ma[4]

[4] *Plant Science Department, South Dakota State University, Brookings, SD, 57007, USA*

Received 19th January 2016
Revised Day Month Year

In this paper, we proposed a link density clustering method for overlapping community detection based on density peaks. We firstly use an extended cosine link distance metric to reflect the relationship of links. Then we introduce a clustering algorithm with fast search for solving the link clustering problem by density peaks with box plot strategy to determine the cluster centres automatically. Finally, we acquire both the link communities and the node communities. Our algorithm is compared with other representative algorithms through substantial experiments on real-world networks. The experimental results show that our algorithm consistently outperforms other algorithms in terms of modularity and coverage.

Keywords: Link community; overlapping community detection; link distance metric; box plot; complex network.

1. Introduction

Overlapping community detection has become a widely discussed topic in the field of complex networks[1]. With the continuous development of research on complex networks, much work has focused on network growth[2] and general statistical mechanics[3], examples of which include community structure in national coauthorship networks and the evolution of interdisciplinary research in Slovenia's

scientific collaboration network[4], the growth and structure of Slovenias scientific collaboration network[5]. Overlapping community detection aims to identify those communities composed of densely connected nodes inside and sparsely connected nodes outside. It has been widely applied in many problems[1−5].

The algorithms for overlapping community detection can be roughly classified as node-based and link-based ones. Traditional overlapping community algorithms are mostly node-based[6−16]. As a representative of node-based algorithms, the CPM (Clique Percolation Method) was proposed by Palla *et al*[6]. However, the CPM algorithm identifies the cliques with restricted structure. Farkas *et al.* extended CPM into weighted networks[7]. Greedy Clique Expansion (GCE) was proposed by Lee *et al.* (2010)[8], and it expanded unique cliques as seeds through greedily optimizing a local fitness function and uses a clean step to merge similar communities identified[8]. Fuzzy c-means clustering algorithm[9] was employed by Zhang *et al.* to detect overlapping communities, and this algorithm combines a novel generalized modularity function based on Q function[10], spectral mapping and fuzzy c-means clustering for identifying fuzzy membership functions[11] in the overlapping community structure. Combining non-negative matrix factorization technique, a popular modular function and a proper feature matrix from diffusion kernel, an algorithm was put forward by Zhang *et al.* (2007) for detecting fuzzy overlapping community structure[11]. In addition, COPRA[12] and SLPA[13] algorithms were proposed based on the idea of label propagation. Most recently, an ant colony based overlapping community detection algorithm was presented by Zhou *et al.* (2015)[14].

Up till now, some research has focused on link community due to the unique advantages of links instead of node communities for overlapping community detection[17,18]. Distinct from the traditional community definition, link communities can be viewed as groups of links. As a representative link-based algorithm, the link clustering (LC) algorithm was proposed by Ahn *et al*[17]. The LC algorithm uses partition density to determine the best communities among the clustering results. It makes the communities identified smaller than the ground-truth. Cazabet *et al.*[18] put forward an algorithm for detecting both static and temporal communities based on links.

Traditional clustering algorithms have been applied to overlapping community detection[17,19,20,21]. Hierarchical clustering algorithm[19] is a typical example used in the LC algorithm. Based on the idea of DBScan[20], LinkScan algorithm is proposed by Sungsu Lim *et al*[21]. The clustering algorithm by fast search and find of density peaks (FSC) was proposed by Alex Rodriguez *et al.*[22], and it has its own advantage for identifying the cluster centres whose densities are higher than their neighbours and whose distances form points with higher densities are also relatively large. However, hierarchical clustering algorithm sometimes finds a local optimal solution. And DBScan algorithm tends to identify dense clusters.

Considering the above issues, we propose to introduce the FSC algorithm for solving the overlapping community detection problem. While the cluster centres of

the FSC algorithm are often determined intuitively, in this research we employ the box plot model to automatically select the cluster centres. Based on the concept of link communities, we put forward a link density clustering algorithm (LDC) for overlapping community detection. Different from the related previous work based on nodes[14,15,16], our algorithm is link based. We present a few key contributions as follows:

(1) We propose an extended cosine link distance metric to evaluate the strength of link relations. This extended link distance metric considers the relationship of links with common neighbours.

(2) We introduce the clustering algorithm by fast search and find of density peaks (FSC)[22] into the link community detection and then use the box plot strategy to obtain cluster centres automatically. The algorithm avoids the process of determining the community detection results by tuning parameters and evaluating the metrics used. Through the extended cosine link distance metric, FSC algorithm can be applied to effectively solve the problem of overlapping community detection.

(3) Experiments on real world networks demonstrate the good performance of our LDC algorithm.

In the rest of this paper, some related algorithms are discussed in Section 2. Our link density clustering algorithm is presented in Section 3. Experiments on complex networks are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Algorithms

In this section, the CPM algorithm (Clique Percolation Method)[6], the LC algorithm (Link Clustering algorithm)[17] and the iLCD algorithm (intrinsic Longitudinal Community Detection algorithm)[18] are presented for comparison with our proposed LDC algorithm. The box plot model is also introduced for automatically selecting the centres of the FSC algorithm.

2.1. *The CPM, LC and iLCD Algorithms*

The CPM algorithm (Clique Percolation Method) was proposed by Palla *et al.* for uncovering the existence of overlapping community structure[6]. The CPM algorithm is based on the assumption that a community is a series of adjacent k-cliques and thus called a k-clique community. The two k-cliques are adjacent if they have k-1 common nodes. Given the clique size k, the CPM algorithm searches all k-cliques in the network. The CPM algorithm detects the k-clique communities among the k-cliques identified based on the concept of the k-clique community. The software CFinder is designed based on the CPM algorithm[23].

The Link Clustering algorithm (LC) was proposed by Ahn *et al.* for revealing the inherent advantage of link community[17]. The LC algorithm firstly calculates the link similarity matrix according to the adjacency matrix and then uses the hierarchical clustering algorithm[19] on the link similarity matrix. Then the link communities are

identified with the highest partition density value. The R package linkcomm was published by Alex et al.[24] based on the LC algorithm[17].

The iLCD algorithm was proposed by Remy Cazabet *et al.* for overlapping community detection on both static and dynamic networks[18]. It is based on the assumption that communities are defined locally[18]. The iLCD algorithm is composed of three steps: updating existing communities, creating a new community, and merging similar communities. In the updating step, a node is added into a community if the mean number of its second neighbours is larger than the estimation of the mean number of its robust second neighbours. In the creation step, if a new edge is added to the network and a minimal community is formed, this minimal community will be created as a new community. The minimal community used in the iLCD algorithm is a clique of three or four nodes. A merging process is performed to merge the two communities when the ratio of nodes in common is higher than the threshold. The threshold set in the iLCD algorithm is (0.2, 0.3) or (0.7, 0.9). The source code of the iLCD algorithm is available at Ref.25.

2.2. *The FSC Algorithm*

The clustering algorithm by fast search and find of density peaks (FSC) was proposed by Rodriguez and Laio (2014)[22]. It is a fast search clustering algorithm[22], and it avoids the process of tuning parameters and the calculation of the evaluation metrics. The FSC algorithm can find the clusters that have higher density in clusters and lower density among the nodes between clusters.

The detailed steps of the FSC algorithm are shown in Algorithm 1 FSC table.

Through identifying the local density and distance between points, the FSC algorithm can find those cluster centres are surrounded by neighbours with local density. Furthermore, the neighbours have a relatively large distance with all those points with a higher local density.

2.3. *The Box Plot Model*

The box plot model has been widely used in descriptive statistics and exploratory data analysis for depicting groups of numerical data through their quartiles. As shown in Fig.1, the box plot model is non-parametric and it shows the maximum value, 75th percentile, 50th percentile, 25th percentile and the minimum value of the statistical distribution. In addition, IQR (interquartile range) is equal to the difference between the 75th percentile and the 25th percentile. In this research we use the box plot model for automatically choosing the centers of link communities in our FSC algorithm with an aim to obtain better results.

3. Our LDC Algorithm

Our proposed LDC algorithm consists of two main steps. First, the extended cosine link distance metric is used to evaluate the relationships of links and the distance

Algorithm 1 FSC

Input: the link distance matrix;

Output: the link communities;

1: For each link i, the local density and the minimum distance are calculated through the distance matrix and Formulas (1)-(3). The local density of link i is defined as:

$$\rho_i = \sum_j f(d_{ij} - d_c). \tag{1}$$

In Formula (1), ρ_i is equal to the number of points that are closer than d_c to point i. d_c is the cutoff distance and function f is defined in Formula (2).

$$f(d_{ij} - d_c) = \begin{cases} 1 \; if \; d_{ij} < d_c \\ 0 \; otherwise \end{cases} \tag{2}$$

If we use δ_i to represent the minimum distance between point i and any other points with higher densities, δ_i is defined in Formula (3):

$$\delta_i = \min_{j:\rho_j > \rho i}(d_{ij}). \tag{3}$$

2: Choose the community centres where the local density and the minimum distance are both relatively higher than their neighbours. According to the centres, the links are classified by their nearest neighbour with higher density values. Finally, the link communities are identified.

matrix of links is calculated. Then, the FSC algorithm[22] is used to identify the dense groups of links according to the distance matrix of links.

3.1. *The Extended Cosine Link Distance Metric*

Before introducing the extended cosine link distance metric, the following symbols are defined in Table 1 for the ease of description.

Table 1. A Summary of Symbols.

Symbol	Description
G	The given network
V	The node set of G
E	The link set of G
A	The adjacency matrix of G
D	Link distance matrix
m	The number of links in G
$n_+(a)$	The neighbours of node a including node a itself
n	The number of nodes in G

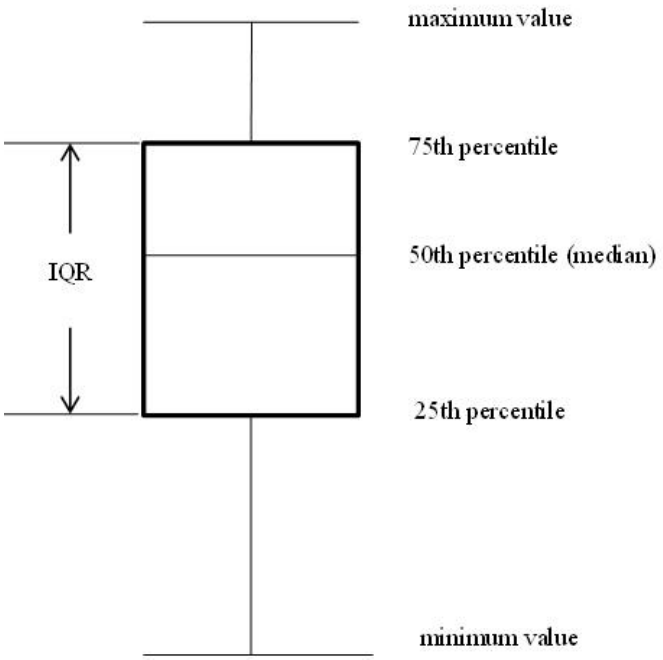


Fig. 1. The Diagram of Box Plot Model.

To introduce our extended cosine link distance metric, we first give the original cosine similarity measure in Formula (4). Formula (4) is defined to calculate the distance between node i and node j.

$$LS(i,j) = \frac{|\, n_+(i) \bigcap n_+(j) \,|}{|\, n_+(i) \bigcup n_+(j) \,|}. \tag{4}$$

The main idea of our extended cosine link distance metric is to employ the link relationship as the distance of links. If two links are more dissimilar, then the distance of the two links is lager. The link distance algorithm is based on the idea of cosine similarity. For evaluating the relationship of links more precisely, the neighbors of the nodes connected by the links are considered. For two given links i and j, link i connects nodes a and b. Link j connects nodes c and d. If link i connects link j, it means that they have one common node. If links i and j have one neighbour node, it means that the nodes connected by the links have one common neighbour node. The extended cosine link distance of link i and link j is defined in Formula (5).

$$D(i,j) = 1 - LS(i,j). \tag{5}$$

We use Formula (5) to calculate the distance between link i and link j, and in this formula, $LS(i,j)$ represents the similarity between link i and link j, as defined below in Formula (6).

$$LS(i,j) = \frac{f(a,c) + f(a,d) + f(b,c) + f(b,d)}{g(a,c) + g(a,d) + g(b,c) + g(b,d)}. \tag{6}$$

$$f(a,c) = \mid n_+(a) \bigcap n_+(c) \mid . \tag{7}$$

$$g(a,c) = \sqrt{\mid n_+(a) \times n_+(c) \mid}. \tag{8}$$

In Formula (7), $n_+(a)$ represents the number of neighbours of node a including node a itself. $f(a,c)$ represents the number of common neighbours of nodes a and c. In Formula(8), $g(a,c)$ represents the product of neighbours of nodes a and c. The numerator of Formula (6) calculates the number of common neighbours between the nodes connected by links i and j; while the denominator of Formula (6) calculates the sum of the product of neighbours of the nodes connected by links i and j. The extended cosine link distance metric thus accommodates the relationship of the links having common neighbour nodes, which is a further extension to the Jaccard similarity used in the LC algorithm as more topological information is incorporated in this metric.

3.2. *Link Density Clustering Algorithm*

Our Link Density Clustering Algorithm (LDC) combines the extended cosine link distance metric and the FSC algorithm[22]. LDC algorithm firstly uses the link distance algorithm to calculate the distance matrix. Then, the community centres and the link communities are determined by the FSC algorithm. Finally, the link communities are transformed into the node communities.

Algorithm 2 LDC

Input: the link set E of graph G;
Output: overlapping communities of graph G;

1: Construct the adjacency matrix A according to the link set E.
2: Based on the extended cosine link distance metric (see Section 3.1), calculate the link distance matrix D_{link}.
3: Use the FSC algorithm (see Section 2) with the link distance matrix D_{link}.
4: Transform the link communities back to the node communities.

The LDC algorithm identifies the link centers and then detects the link communities according to the community centres. Firstly, we assign indexes to the links.

Secondly, the extended cosine link distance metric is applied to the links (see Section 3.1) and the link distance matrix is obtained. Thirdly, the FSC algorithm is used on the link distance matrix (see Section 2.2) for identifying the link communities. Finally, the link communities are transformed back into node communities according to their indexes.

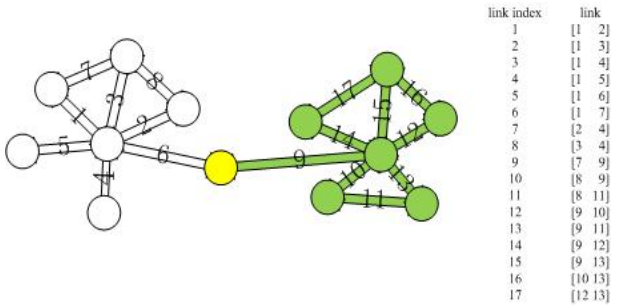
The strategy of automatically choosing centres of the link communities in the FSC algorithm is employed from the box plot model presented in Section 2.3. The strategy is to identify the links with the value of ρ being higher than 75th percentile and the value of δ being higher than the sum between 75th percentile and 1.5 times of IQR. If less than two link centers are found by using the above strategy, the strategy is then adjusted to identifying the links with δ being higher than 75th percentile.

The time complexity of our extended cosine link similarity is $O(m^2 + n^2)$ and the corresponding space complexity is $O(m^2 + n^2)$. The time complexity of the FSC algorithm is $O(m^2)$ and the corresponding space complexity is $O(m^2)$. Hence, the time complexity of our LDC algorithm is $O(m^2 + n^2)$ and the space complexity of LDC algorithm is $O(m^2 + n^2)$.

An example for illustrating our LDC algorithm is shown in Fig.2. In this example network, there are 12 nodes and 17 links. And it naturally forms two obvious communities and the links after sorting with their indexes are shown in Fig.2 (a). In the LDC algorithm, the link distance matrix is shown in Fig.2 (b). Combined with the indexes of the links in Fig.2(a), the link set {1,2,3,4,5,6,7,8} forms an obvious community and another link set {9,10,11,12,13,14,15,16,17} forms another natural community. Our link distance matrix also shows the two blocks corresponding to the two link communities in Fig.2 (b). The links in the same community are closer than those outside. As shown in Fig.2 (c), the cluster centres chosen by LDC are $e_{1,4}$ (link 3) and $e_{9,13}$ (link 15), which are shown as two overlapping brown color points in the top right corner. In Fig.2 (c), the red line represents the 75th percentile of ρ and the green line represents the sum of the 75th percentile and 1.5 times IQR of δ. The community cluster centre $e_{1,4}$ (link 3) is colored white and the community of cluster centre $e_{9,13}$ (link 15) is colored green. The overlapping node of the two communities is node 7 (in yellow).

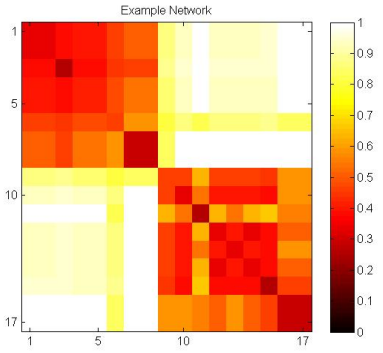
4. Experiments

In this section, the performance of LDC is extensively tested on several real-world networks. In all the experiments, the cutoff distance in our LDC algorithm we take is the same as the original FSC algorithm and it sets up to the two percent of the total distance matrix because of the FSC algorithm is robust to the choice of the cutoff distance. We compared our algorithm with other three algorithms: CPM[6], LC[17], and iLCD[18].

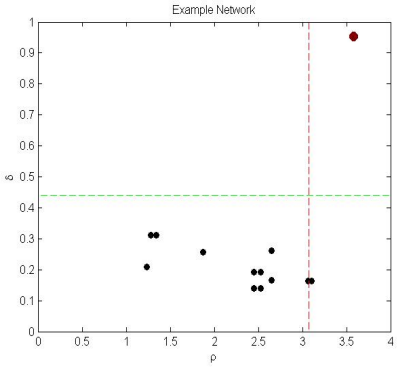


link index	link
1	[1 2]
2	[1 3]
3	[1 4]
4	[1 5]
5	[1 6]
6	[1 7]
7	[2 4]
8	[3 4]
9	[7 9]
10	[8 9]
11	[8 11]
12	[9 10]
13	[9 11]
14	[9 12]
15	[9 13]
16	[10 13]
17	[12 13]

(a) Distance matrix



(b) Decision graph



(c) Decision graph

Fig. 2. Results of *LDC* algorithm on the example Network

4.1. *Data Source*

The real-world networks tested by our LDC algorithm are classified as empirical networks and unempirical networks. The empirical networks represent the real-world networks with ground-truth. The empirical networks we used are the Karate network[26], the Dolphin network[27], and the Politics network[28]. The empirical networks represent the real-world networks without ground-truth. The unempirical network is the Power grid network[29]. The detailed parameters of these networks are shown in Table 2.

Table 2. Real-world networks.

Dataset	Nodes	Links	Average Degree	Community Number (CN)
Karate[26]	34	78	4.59	2
Dolphin[27]	62	159	5.13	2
Politics[28]	105	441	8.4	3
Power grid[29]	4941	6954	2.9	-

4.2. *Evaluation Metrics*

Evaluation metrics develop with the complex network research. There are many evaluation metrics proposed for evaluating overlapping communities, such as Extended Modularity (EQ)[30], Clustering Coverage (CC)[21], Modularity Density[31], Community Numbers (CN)[14], Normalized Mutual Information (NMI)[32], and Partition Density[17]. To evaluate the performance of the above algorithms on different types of networks, the Extended Modularity (EQ)[30], Clustering Coverage (CC)[21] and Community Numbers (CN)[14] are adopted for all networks as three commonly used metrics in the evaluation of overlapping community[33]. Meanwhile, an extended Normalized Mutual Information (NMI) is used for the empirical real-world networks also as a commonly used metric[33].

Extended Modularity (EQ) evaluation metric is an extension of modularity[28] for overlapping community detection, and is proposed by Shen *et al.*[30]. EQ is widely used for the evaluation of the overlapping community results. The calculation of EQ is shown in Formula (9).

$$EQ = \frac{1}{2m}\sum_{l=1}^{C_l}\sum_{i\in l, j\in l}\frac{1}{O(i)O(j)}(A(i,j) - \frac{k(i)k(j)}{2m}). \tag{9}$$

In Formula (9), C_i is one of the node communities and the network is divided into C_l communities. $O(i)$ represents the number of communities that node i belongs to. The meaning of other symbols is given in Table 1. If the link communities have a high EQ value, it means that the community has a good modularity.

Lancichinetti *et al.*[32] proposed an extension of Normalized Mutual Information (NMI), and it has become one of the most popular evaluation metrics for evaluating overlapping communities. NMI has to be provided with the ground-truth, and the range of NMI is from 0 to 1. The higher the value of NMI, the closer the result is to the actual communities.

Clustering Coverage (CC)[21] is used for evaluating nodes covered by algorithms, and we use Formula (10) to calculate CC. In Formula (10), the number of nodes detected by algorithms is denoted as n_1 and the total number of nodes in network is denoted as n. CC is the percentage of n_1 divided by n. The higher the value of CC , the more nodes the algorithm covers.

$$CC = \frac{100 \times n_1}{n}. \tag{10}$$

Community Numbers (CN)[14] is used for measuring the difference between the community numbers by different algorithms and the ground-truth community numbers in empirical networks. CN is also a reference evaluation metrics in unempirical networks. The closer of CN is to the ground-truth, the better the result is.

4.3. *Experimental Results*

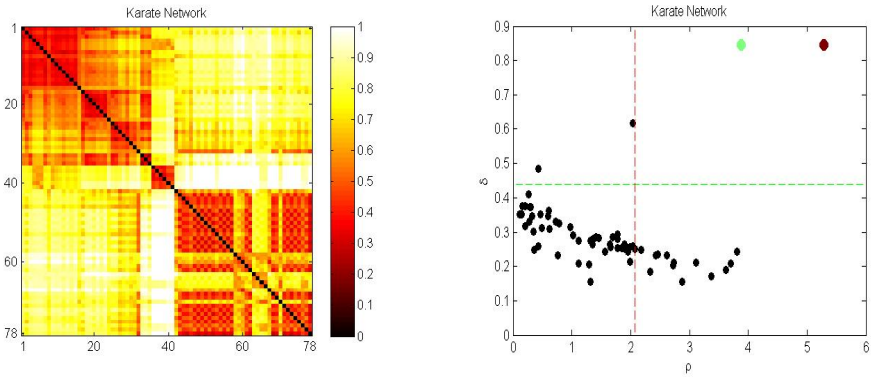
4.3.1. *Results on Empirical Real-World Networks*

In Table 3, LDC obtained two communities with the best *EQ* value of 0.276, the best *NMI* value of 0.556 and covered the whole nodes in the network. And there are two clear blocks in Fig.3(a). Meanwhile, Fig. 3(b) shows the decision process of choosing the clusters centres. The LC algorithm obtained the suboptimal result on the *EQ*, *NMI* and *CC* values. However, the *CN* is far from the ground-truth given in Table 2. The result identified by the CPM algorithm has higher *EQ*, *NMI* and *CC* values than those obtained from the iLCD algorithm.

Table 3. Performance Comparison on Karate network[a].

Algorithm	EQ	NMI	CN	CC(%)
LDC	0.276	0.556	2	100.00
LC	0.260	0.309	8	97.06
CPM	0.186	0.175	3	94.12
iLCD	0.141	0.192	12	73.53

[a] **The data marked in bold are the best values among all algorithms.**



(a) Distance matrix (b) Decision graph

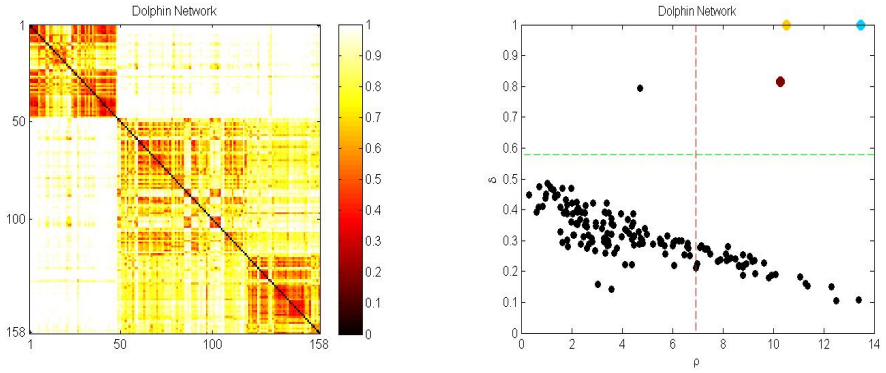
Fig. 3. Results of *LDC* algorithm on Karate Network

In Table 4, LDC algorithm identified four communities with the best *EQ* value of 0.379 and the best *NMI* value of 0.478 and covered all nodes in Dolphin Network. The distance matrix and the decision graph of LDC algorithm are shown in Fig.4 (a) and (b). The *EQ* and *NMI* values obtained from the LDC algorithm are both higher than those obtained from the CPM algorithm. LC generated 13 communities with an *EQ* value of 0.261 and the *NMI* value of 0.146. And the *EQ* value of the result obtained by LC is worse than other algorithms.

Table 4. Performance Comparison on Dolphin Network[b].

Algorithm	EQ	NMI	CN	CC(%)
LDC	0.379	0.478	3	100.00
LC	0.261	0.146	13	67.74
CPM	0.361	0.318	4	74.19
iLCD	0.277	0.141	13	70.97

[b] **The data marked in bold are the best values among all algorithm.**



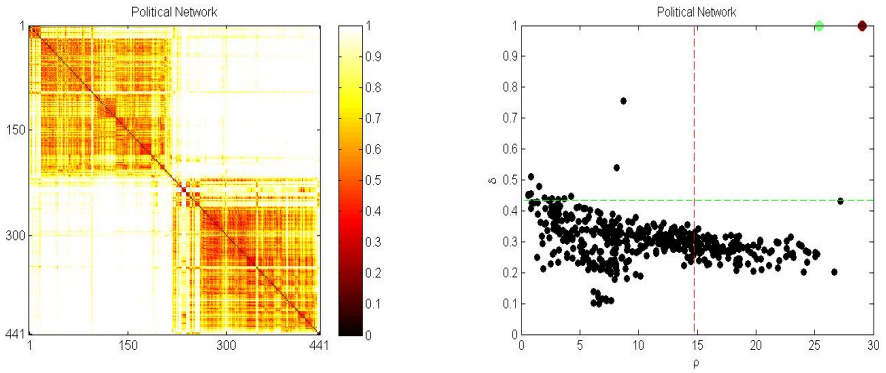
(a) Distance matrix (b) Decision graph

Fig. 4. Results of *LDC* algorithm on Dolphin Network

Table 5. Performance Comparison on Political Network[c].

Algorithm	EQ	NMI	CN	CC(%)
LDC	0.430	0.358	2	100.00
LC	0.176	0.102	32	88.57
CPM	0.437	0.247	4	99.05
iLCD	0.175	0.086	41	95.24

[c] **The data marked in bold are the best values among all algorithms.**



(a) Distance matrix (b) Decision graph

Fig. 5. Results of *LDC* algorithm on Political Network

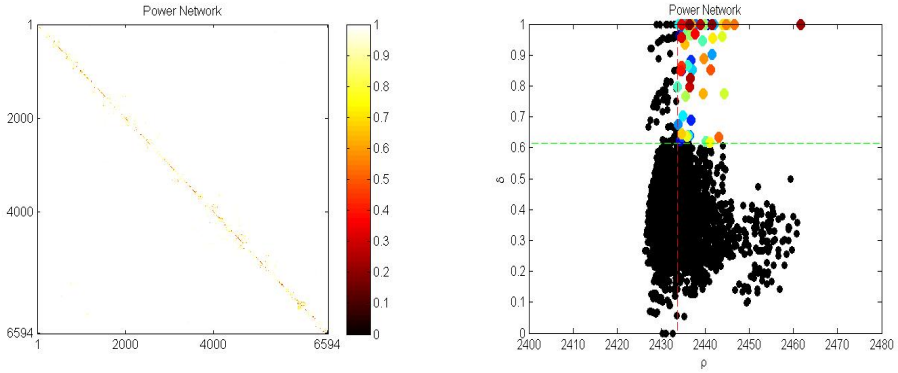
In Table 5, three communities with an *EQ* value of 0.430 and *NMI* value of 0.284 found by LDC have covered all nodes in Political Network. And the *NMI* value and CC values obtained by LDC are better than CPM. And the *EQ* value of the result obtained by LDC is slightly lower than CPM. The distance matrix and the decision graph of LDC are shown in Fig.5 (a) and (b). Four communities identified by CPM with an *EQ* value of 0.437 and *NMI* value of 0.247 covered 99.05% of all nodes in the Political Network and were all better than both LC and iLCD. 41 communities obtained by iLCD have an *EQ* value of 0.175, an *NMI* value of 0.086, and a *CC* value of 95.24%. These values obtained by iLCD exceed the LC algorithm except the *CC* value.

4.3.2. *Results on the Unempirical Real-World Network*

Table 6. Performance Comparison on Power Grid Network[d].

Algorithm	EQ	CN	CC(%)
LDC	0.847	63	97.11
LC	0.199	375	26.86
CPM	0.158	297	19.25
iLCD	0.127	434	19.11

[d] The data marked in bold are the best values among all algorithms.



(a) Distance matrix (b) Decision graph

Fig. 6. Results of *LDC* algorithm on Power Network

In Table 6, 63 communities were found by LDC with the best *EQ* value of 0.847 and included 97.11% of all nodes in Power Grid network. The distance matrix is shown in Fig. 6(a) and the decision process of choosing the cluster centres is shown in Fig. 7(b). 375 communities found by LC with an *EQ* value of 0.199 and with a coverage value of 26.86%, and these results are better than those obtained

from CPM. 297 communities identified by CPM with an *EQ* value of 0.158 covered 19.25% of all nodes. 434 communities found by iLCD with an *EQ* value of 0.127 and a *CC* value of 19.11%, which are worse than all the other algorithms.

After the extensive experiments on both empirical and unempirical real-world networks, we find that our LDC algorithm can identify communities with relative higher *EQ* and *NMI* values. Meanwhile, it covers more nodes in the networks compared with other three representative algorithms.

5. Conclusions and Future Work

In this paper, a link density clustering (LDC) algorithm for overlapping community detection is presented. LDC first calculates the distance matrix through the extended cosine link distance metric, which we propose to make a better use of the topological information of the links. Then, the FSC clustering algorithm is used on the distance matrix and clusters the links into link communities. Finally, the node communities are detected according to the link communities. Experimental results on two types of complex networks demonstrated the good performance of our LDC algorithm in terms of both the *EQ* and *CC* evaluation metrics.

In our future work, we will explore both the further development of LDC and the application potential of LDC in problems of various domains. In terms of further algorithm development, we will focus on refining the process of choosing centre links with an aim to improve the performance of LDC on synthetic networks. In addition, we will also consider how to further reduce the computational complexity of LDC so that it can be used for large-scale networks. In terms of real-world applications, we plan to investigate the potential of LDC when applied to more real-world complex networks in various domains. This includes employing LDC to enhance the performance of social network based recommender systems[34,35], better analyse sensor networks[36–38], biological networks[39,40], and in particular, better clustering information networks[41,42].

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61472159, 61572227), Development Project of Jilin Province of China (Nos. 20130522111JH, 20140101180JC) and Jilin Province Development and Reform Commission (No. 2014N143). This work is also supported in part by the Biochemical Spatiotemporal Network Resource Centre (3SP680) of South Dakota State University. WP is supported by the PECE bursary provided by Scottish Informatics and Computer Science Alliance.

References

1. S. Fortunato, *Physics reports*, **486**(3), 75 (2010).
2. M. Perc, *Journal of The Royal Society Interface*, **11**(98), 20140378 (2014).

3. A. Van De Walle and G. Ceder, *Reviews of Modern Physics*, **74**(1), 11 (2002).
4. B. Luzar, Z. Levnajic, J. Povh and M. Perc, *Plos one*, **9**(4), e94429 (2014).
5. M. Perc, *Journal of Informetrics* **4**(4), 475 (2010).
6. G. Palla I. Derenyi, I. Farkas and T. Vicsek, *Nature* **435**, 814 (2005).
7. I. Farkas, D. Abel, G. Palla and T. Vicsek, *New Journal of Physics* **9**, 180 (2007).
8. C. Lee, F. Reid, A. McDaid and N. Hurley, *arXiv preprint arXiv* **1002**, 1827 (2010).
9. S. Zhang, R.S. Wang and X.S. Zhang, *Physica A: Statistical Mechanics and its Applications* **374**, 483 (2007).
10. M.E. Newman and M. Girvan, *Physical Review E* **69**, 026113 (2004).
11. S. Zhang, R.S. Wang and X.S. Zhang, *Physical Review E* **76**, 046103 (2007).
12. S. Gregory, *New Journal of Physics* **12**, 103018 (2010).
13. J. Xie and B.K. Szymanski, Community detection using a neighborhood strength driven label propagation algorithm, in *Network Science Workshop (NSW), 2011 IEEE* (2011) pp.188-195.
14. X. Zhou, Y. Liu, J. Zhang, T. Liu and D. Zhang, *Physica A: Statistical Mechanics and its Applications* **427**, 289 (2015) .
15. P. Fu, S. Zhu, A. Zhu and X. Dong, *International Journal Of Modern Physics B* **28**, 1450039 (2014).
16. X. Zhang, S. Fei, C. Song, X. Tian and Y. Ao, *International Journal Of Modern Physics B* **29**, 1550029 (2015).
17. Y.–Y. Ahn, J.P. Bagrow and S. Lehmann, *Nature* **446**, 761 (2010).
18. R. Cazabet, F. Amblard and C. Hanachi, in *Social Computing (SocialCom), 2010 IEEE Second International Conference on* (2010) pp.309-314.
19. M.R. Anderberg, Cluster Analysis for Applications: Probability and Mathematical Statistics: A Series of Monographs and Textbooks, *Academic Press* (2014).
20. M. Ester, H.P. Kriegel, J. Sander and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proc. 2nd ACM SIGKDD Intl Conf. on Knowledge Discovery and Data Mining* (1996) **96** (34) pp.226–231.
21. S. Lim, S. Ryu, S. Kwon, K. Jung and J.-G. Lee, LinkSCAN*: Overlapping community detection using the link-space transformation, in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on* (2014) pp.292-303.
22. A. Rodriguez and A. Laio, *Science* **344** (6191), 1492 (2014).
23. http://www.cfinder.org last retrieved: January 12, 2016.
24. A.T. Kalinka and P. Tomancak, *Bioinformatics* **27**, 14 (2011).
25. http://cazabetremy.fr/iLCD.html last retrieved: January 12, 2016.
26. W.W. Zachary, *Journal of anthropological research* **33** (4) 452 (1977).
27. D. Lusseau *et al.*, *Behavioral Ecology and Sociobiology* **54** (2003) pp.396–405.
28. M.E. Newman, *Proceedings of the National Academy of Sciences* **103**, 8577 (2006).
29. D.J. Watts and S.H. Strogatz, *Nature* **393** 440 (1998).
30. H. Shen, X. Cheng, K. Cai, and M.-B Hu, *Physica A: Statistical Mechanics and its Applications* **388**, 1706 (2009).
31. Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang and L. Chen, *Physical review E* **77**, 036109 (2008).
32. A. Lancichinetti, S. Fortunato and F. Radicchi, *Physical review E* **78**, 046110 (2008).
33. J. Xie, S. Kelley and B. K. Szymanski, *ACM Computing Surveys* **45**(4), 43 (2013) p.43.
34. C. Luo, W. Pang and Z. Wang, Hete-CF: Social-Based Collaborative Filtering Recommendation using Heterogeneous Relations, in *Data Mining (ICDM), 2014 IEEE International Conference on* (2014) pp.917–922.
35. T. Ma, J. Zhou, M. Tang and Y. Tian, *IEICE TRANSACTIONS on Information and*

Systems **98**(4), 902 (2015).
36. J. Shen, H. Tan, J. Wang, J.W., Wang and S.Y. Lee, *Journal of Internet Technology* **16**(1), 171 (2015).
37. S. Xie and Y.X. Wang, *Wireless Personal Communications* **78**(1), 231 (2014).
38. P. Guo, J. Wang, B. Li and S.Y. Lee, *Journal of Internet Technology* **15**(6), 929 (2014).
39. A. Gavin *et al.*, *Nature* **440**, 631 (2006).
40. N. Krogan *et al.*, *Nature* **440**, 637 (2006).
41. Y. Sun and J.W. Han, *Synthesis Lectures on Data Mining and Knowledge Discovery* **3**(2), 1 (2012).
42. C. Luo, W. Pang and Z. Wang, in *Advances in Knowledge Discovery and Data Mining. Springer International Publishing* 548 (2014).