Expert Systems With Applications 69 (2017) 40–49

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Detecting outlier pairs in complex network based on link structure and semantic relationship

L. Liu [a], W.L. Zuo [a,b], T. Peng [a,b,*]

[a] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[b] Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Changchun 130012, China

ARTICLE INFO

Article history:
Received 23 April 2016
Revised 10 October 2016
Accepted 12 October 2016
Available online 15 October 2016

Keywords:
Outlier pair detection
Complex network
Link structure
Semantic relationship
K-step index

ABSTRACT

In this paper, we propose an outlier pair detection method, called LSOutPair, which discovers the vast differences between link structure and semantic relationship. LSOutPair addresses three important challenges: (1) how can we measure the target object's link similarity among multi-typed objects and multi-typed relations? (2) how can we measure the semantic similarity using the short texts? (3) how can we find the objects' maximum differences between link structure and semantic relationship? To tackle these challenges, LSOutPair applies three main techniques: (1) two matrices are used to store link similarity and semantic similarity, (2) a *k*-step index algorithm, which calculates the term weighting for each object, (3) applying the linear transformation of Frobenius norm to matrices can obtain the top-K outlier pairs. LSOutPair considers link and semantics in complex network simultaneously, which is a new attempt in data mining. Substantial experiments show that LSOutPair is very effective for outlier pair detection.

1. Introduction

When analyzing multi-typed objects and multi-typed relations in complex information networks, identifying rare, interesting and outstanding objects, patterns or subgraphs is at least, or even more meaningful and significant than knowing the common data distribution or models. Outlier detection, as an important branch in the field of data mining, can be applied to extract the objects, patterns or subgraphs which significantly deviate from others in the network. One of the widely accepted definition of outlier is provided by Hawkins (1980): An outlier is an observation that deviates so much from other observations as to arouse suspicion that it is generated by a different mechanism. Most traditional outlier detection methods are only applicable to single-typed networks. When it comes to complex networks (Gupta, Gao, Sun, & Han, 2012a; Gupta, Gao, Sun, & Han, 2012b; Dalmia, Gupta, & Varma, 2015; Gupta, Mallya, Roy, Cho, & Han, 2014), there are multi-typed objects instead of single-typed objects and each type of objects can be an outlier. For example, in bibliographic network, either an author or a research area can be an outlier.

Quite different from most existing work that identifying outliers in networks (Aggarwal & Sathe, 2015; Angiulli & Fassetti, 2016; Dufrenois & Noyer, 2016; Fusters et al., 2013; Maervoet, Vens, Berghe, Blockeel, & DeCausmaecker, 2012), our proposed work aims at discovering outlier pairs. Compared with the definition of outlier, we provide a definition of outlier pair. An outlier pair is two specific objects whose link structure similarity deviates so much from semantic relationship similarity as to arouse suspicion that they are generated by a different mechanism. When mining information from complex networks, it is essential to consider the link relation between objects and capture their semantic relations. How about there is vast difference of two objects between link structure and semantic relationship? Either of the two objects in an outlier pair may be normal from the perspective of link structure or semantic relationship. The two objects are suspicious when their structure similarity and semantic similarity are quite different. For example, when detecting communities or extracting domain experts, we probably treat link structure and the contents in the network as the major considerations in the process of data analysis. If two authors often collaborate on a paper and their research areas are almost the same, the two authors, of course, can be partitioned to the same community. Likewise, if two actors often co-star in a movie, and the film types, that the actors are good at, are similar, then the two actors can be classified into the same group. However, once there is vast difference between link structure similarity and semantic relationship similarity, say, two authors often collaborate on a paper but their research areas are totally different, at this time, the two authors can be regarded as an outlier pair, instead of an outlier. Perhaps two authors' research areas are almost the same but they never collaborate on a paper, then we can recommend them to each other to see whether they can carry out

* Corresponding author.
E-mail addresses: liulu12@mails.jlu.edu.cn (L. Liu), wanlizuo@hotmail.com (W.L. Zuo), tpeng@jlu.edu.cn (T. Peng).

http://dx.doi.org/10.1016/j.eswa.2016.10.026
0957-4174/

academic exchanges. The two authors are also regarded as an outlier pair.

Motivated by the above ideas, we propose an outlier pair detection method in complex network. We first formalize the outlier pair detection problem w.r.t the vast differences between link structure and semantic relationship. Then, we introduce a novel notion of link structure model to measure the proximity among objects based on their structural distribution. To calculate the semantic similarity among objects, a *k*-step index algorithm is proposed to obtain the term weighting for each object. Finally, the Frobenius norm of a matrix and linear transformation are combined to rank the outlierness of objects in the complex networks. We conduct several experiments on AMiner and Yahoo!Movies to verify the effectiveness of the proposed LSOutPair based on link structure and semantic relationship. The experimental results show that our proposed LSOutPair can achieve high accuracy.

The contributions of this paper are summarized as follows.

(1) We introduce the notion of outlier pair detection in complex network by combining the link structure and semantic relationship among objects.
(2) The concept called *k*-step index is defined to calculate the term weighting of the target objects in the complex network.
(3) We propose an LSModel to rank the candidates' differences between link structure and semantic relationship and obtain the top-K outlier pairs.
(4) Extensive experiments on two real datasets demonstrate the effectiveness of our proposed method.

The rest of this paper is organized as follows. We discuss the related work in Section 2. Section 3 presents the definitions and concepts used in this paper. The overall framework of our proposed LSOutPair is also illustrated in Section 3. We discuss the link structure model and semantic relationship model in Section 4. Section 5 describes how to detect the outlier pairs using the LSModel. Several comprehensive experiments are performed to evaluate the effectiveness and efficiency of our proposed method in Section 6 in which the experiment settings, performance metrics, datasets and results are all provided. Section 7 discusses the main advantages and limitations of LSOutPair in comparison with the existing methods. Section 8 draws the conclusions.

2. Related work

Outlier detection plays an important role in complex network. Most existing work discovers outliers in the form of single vertex or subgraphs in such networks. In what follows, we give an overview of previous work on outlier detection and its applications.

Mining outlier of single vertex. For this kind of outliers, their attributes or community distribution may be different from others. The following techniques can be used to detect outliers in form of single vertex. Yao, Mark, and Rabbat (2012) described an anomaly detection method combining PageRank algorithm and proximity graphs. They utilized a variant of the PageRank algorithm to generate an outlier score and judge whether each data point was anomalous. Rossi, Gallagher, Neville, and Henderson (2013) proposed a dynamic outlier detection approach which studied the temporal behavior of nodes in the graph. According to the temporal behaviors of individual nodes, this model can detect anomalous transitions and interesting patterns. Gupta, Gao, Aggarwal, and Han (2013a) defined a novel concept called Community Distribution Outliers (*CDOutliers*) which used non-negative matrix factorization to detect objects whose community distribution did not follow other popular community distribution patterns. They also extracted outliers in the form of single vertex in heterogeneous networks. Cao, Wei, Yang, and Rundensteiner (2015) presented an online outlier exploration platform, called ONION, that could detect individual outliers over large datasets.

Mining outlier of subgraphs. There are about two kinds of outliers in the form of subgraphs. The one is that any one of objects in the subgraph is a normal object but multiple objects showing the same property can be regarded as outliers. The other one is the outliers in a subgraph can be affected by the connectivity structure. The following techniques can be applied to detect these kinds of outliers. Akoglu, Tong, and Koutra (2014) made a comprehensive survey about graph-based anomaly detection. The survey described lots of the state-of-the-art methods for outlier data represented as graphs. Zhuang et al. (2014) proposed a query-based subnetwork outlier detection method for heterogeneous networks. They defined the notion of subnetwork similarity and ranked subnetworks according to the outlierness. The outliers are represented by subgraphs.

There are also other forms of outliers such as outlier patterns (Dai, Zhu, Lim, & Pang, 2015), outlier correlations (Karppa, Kaski, & Kohonen, 2016), video anomaly detection (Zhang, Lu, Zhang, & Ruan, 2016) and so on. Dai et al. (2015) defined extreme rank anomaly collection (ERAC) to discover objects exhibiting similar extreme behaviors. It discovered top-K anomalous collections in the datasets. Karppa et al. (2016) proposed a faster subquadratic algorithm to find outlier correlations of vectors. They computed the inner product of vectors to find outlier pairs. Representing multi-typed objects using one vector is not applicable during the process of calculation. Therefore, the algorithm can be used in single-typed networks instead of multi-typed networks. Angiulli, Basta, Lodi, and Sartori (2013) proposed a distributed outlier detection method. The method applied parallel computation to increase the efficiency when performing outlier detection. The experimental results show that their algorithm has good scalability with the increasing number of nodes. In addition, outlier detection has a wide variety of applications such as intrusion detection (Kim, Lee, & Kim, 2014), anomalous maritime trajectory (Lei, 2016), spam detection (Dai et al., 2015; Xie, Wang, Lin, & Yu, 2007), and database activity monitoring (Aydin, Karakose, & Akin, 2015; Kim, Cho, Lee, Kang, & Kim, 2013), and so on.

Although there are many attempts and achievements on outlier detection in the form of single vertex or subgraphs, few of them perform outlier pair detection. The proposed work in this paper detects outlier pairs in complex network combining the link structure and semantic relationship. The vast differences between link structure similarity and semantic similarity can be applied to rank outlierness of outlier pairs in the network.

3. Problem definition

In this section, we start with some formalized problem definitions and some preliminary concepts. Then, we describe the overall framework of our proposed outlier pair detection model.

Definition 1. (*Linear Transformation*) (Leon, 2011). A Mapping M from a vector space X into a vector space Y is said to be a linear transformation if

$$M(\alpha x_1 + \beta x_2) = \alpha\ M(x_1) + \beta M(x_2)$$

for all $x_1, x_2 \in X$ and all scalars α and β.

Definition 2. (*Outlier Pair*). Given an object pair (i, j), the link structure similarity of (i, j) is represented by l_{ij} and the semantic relationship similarity is represented by s_{ij}. An object pair (i, j) is said to be an outlier pair if the difference between l_{ij} and s_{ij}, that is, $|l_{ij} - s_{ij}|$, deviates so much from other object pairs' differences between link structure similarity and semantic relationship similarity.

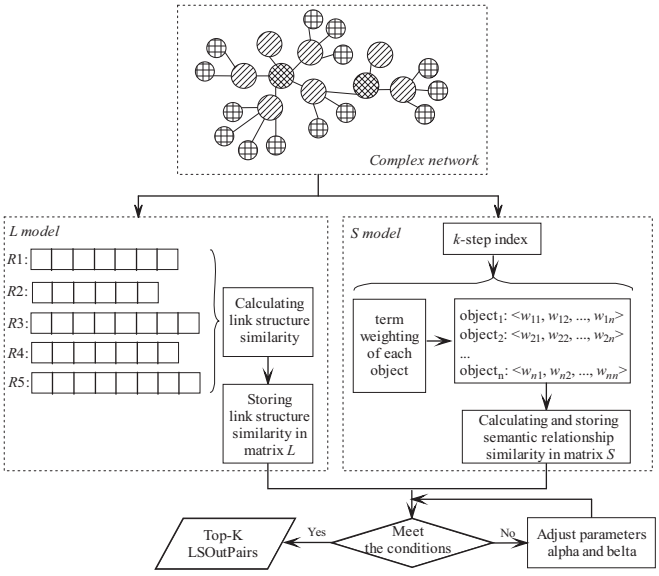


Fig. 1. The overall framework of LSOutPair.

Given n objects in a complex network as input, calculate link structure similarity and semantic relationship similarity, respectively. Find the top-K values of the differences between link structure similarity and semantic relationship similarity of two objects i and j. The top-K pairs (i, j) are considered to be *outlier pairs*.

Definition 3. (*Outlier Pair Matrix*). Assume $L = (l_{ij})$ and $S = (s_{ij})$ are both $n \times n$ matrices representing link structure similarity and semantic relationship similarity, respectively. In addition, $\|\alpha L_{n\times n}\|_F - \|\beta S_{n\times n}\|_F < \varepsilon$. Outlier pair matrix $M_{os} = (m_{ij})_{nn}$ is the absolute value of $\alpha\boldsymbol{L} - \beta\boldsymbol{S}$ whose (i, j) entry is $|l_{ij} - s_{ij}|$ for each ordered pair (i, j). α and β are two scalars which adjust the values in L and S.

For example,

$$\text{If } L_{3\times3} = \begin{bmatrix} 2 & 3 & 7 \\ 9 & 5 & 1 \\ 6 & 4 & 8 \end{bmatrix} \text{ and } S_{3\times3} = \begin{bmatrix} 1 & 5 & 3 \\ 2 & 2 & 12 \\ 1 & 9 & 3 \end{bmatrix} (\alpha = \beta = 1),$$

$$\text{then } M_{os} = \begin{bmatrix} |2-1| & |3-5| & |7-3| \\ |9-2| & |5-2| & |1-12| \\ |6-1| & |4-9| & |8-3| \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 7 & 3 & 11 \\ 5 & 5 & 5 \end{bmatrix}.$$

In the real world datasets, the matrix M_{os} is a symmetric matrix. The values on the diagonal line are zero. Fig. 1 shows the framework of our proposed LSOutPair. We calculate the similarity between objects in complex network from two perspectives. The first one is the link structure similarity between objects, which is shown in the left part of Fig. 1. The second one is the semantic relationship similarity between objects, which is shown in the right part of Fig. 1. Then, the Frobenius norm and linear transformation is combined to get the vast differences between matrices L and S. The top-K outlier pairs can be obtained through adjusting the parameters α and β.

4. Link structure model and semantic relationship model

Different from traditional outlier detection, which discovers objects whose attributes or data distribution are significantly different from others, we aim to detect outlier pairs. The definition *outlier pairs* represent a novel type of outliers considering both the link structure similarity and semantic relationship similarity. In this section, we detail the Link Structure Model and Semantic Relationship Model (LSModel), respectively. The outlier pair detection method using LSModel is devoted in Section 5.

Table 1
An example of ten records covering several authors.

ID	Co-author *ID*	*ID*	Co-author *ID*
P_1	a_1, a_2, a_3, a_4	P_6	a_2, a_8, a_9
P_2	a_1, a_2, a_4, a_5	P_7	a_2, a_4, a_5, a_{10}
P_3	a_4, a_5, a_6, a_7	P_8	a_3, a_6, a_{11}
P_4	a_1, a_2, a_4, a_6	P_9	a_1, a_2, a_3, a_4
P_5	a_2, a_8	P_{10}	a_1, a_3, a_6, a_7

4.1. Link structure model

Generally, multi-typed objects, connected by multi-typed edges representing relations between objects, can form complex information networks. Some researchers delve into the link structure to discover similar or anomalous objects from the whole network. Some other researchers use the content information related to the objects to explore the potential semantic relationship among them. However, once there is a link or a short path between two objects but there is almost no semantic relationship between them, we regard this kind of data pairs as outlier pairs. In what follows, we propose a link structure model to describe the link similarity between objects.

An adjacent square matrix L of length n on a network G consists of target objects and aims to obtain the link similarity according to their connection information. For example, in a bibliographic network, authors can be regarded as target objects. The link similarity between them can be obtained according to the co-author information. In a movie network, actors also can be regarded as target objects. We can get the link similarity between them according to their collaboration information. Assume that there are n authors in a bibliographic network. L is an n by n matrix that describes the link structure similarity between any two authors. Thus,

$$L_{n\times n} = \begin{bmatrix} \boldsymbol{l_1} \\ \boldsymbol{l_2} \\ \vdots \\ \boldsymbol{l_n} \end{bmatrix} = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$

where

$$l_{ij} = \begin{cases} \frac{|\delta_{ij}|}{|\delta_{i.} \cup \delta_{.j}|} + \frac{1}{N} \sum_{r=1}^{n} \frac{|\delta_{ir} \cup \delta_{rj}|}{|\delta_{r.}|} & i \neq j \\ 0 & i = j \end{cases} \quad (1)$$

subject *to the following conditions*

$r \neq i, r \neq j$
$\delta_{ir} \neq \emptyset$ and $\delta_{rj} \neq \emptyset$

δ_{ij} denotes the set of paper *ID* that author a_i and author a_j appear together. $\delta_{i.}$ represents the set of paper *ID* including author a_i. $\delta_{.j}$ represents the set of paper *ID* including author a_j. N is the sum of r that satisfies $\delta_{ir} \neq \emptyset$ and $\delta_{rj} \neq \emptyset$. It can been seen from Eq. (1) that l_{ij} is equal to l_{ji}, so $L_{n\times n}$ is symmetric ($L^T = L$). $\frac{|\delta_{ij}|}{|\delta_{i.} \cup \delta_{.j}|}$ component reflects the importance of direct co-authors. The more papers two authors collaborate on, the higher link structure similarity they have. Then, $\sum_{r=1}^{n} \frac{|\delta_{ir} \cup \delta_{rj}|}{|\delta_{r.}|}$ component measures the number of indirect collaboration between author a_i and author a_j. It reflects the probability of future academic exchanges between the two authors. That is, if a_i and a_j both collaborate with a_r, a_i and a_j will probably carry on academic exchanges and collaborate on a paper sometime. For example, Table 1 lists 10 records representing 10 papers written by several authors. The link structure similarity

Table 2
An example of ten paper ID and the corresponding keywords.

ID	Keywords	ID	Keywords
P_1	A, B, C, H	P_6	G, I, K, L
P_2	C, D, E	P_7	C, F, G, I
P_3	A, C, F, G	P_8	C, D, E, I, J
P_4	C, D, I, J	P_9	A, C, D
P_5	A, G, I	P_{10}	A, B, D, E

between authors a_3 and a_6 is calculated as follows. Since l_{ij} is subject to the two conditions above, $\sum_{r=1}^{n} \frac{|\delta_{ir} \cup \delta_{rj}|}{|\delta_{r\cdot}|}$ component of $l_{3,6}$ contains five parts ($r = 1,2,4,7,11$).

$$l_{3,6} = \frac{\{P_8, P_{10}\}}{\{P_1, P_8, P_9, P_{10}\} \cup \{P_3, P_4, P_8, P_{10}\}} + \frac{1}{5} \times \left\{ \begin{array}{l} \left\{ \frac{\{P_1,P_9,P_{10}\} \cup \{P_4,P_{10}\}}{\{P_1,P_2,P_4,P_9,P_{10}\}} \right\}_{r=1} + \left\{ \frac{\{P_1,P_9\} \cup \{P_4\}}{\{P_1,P_2,P_4,P_5,P_6,P_7,P_9\}} \right\}_{r=2} \\ + \left\{ \frac{\{P_1,P_9\} \cup \{P_3,P_4\}}{\{P_1,P_2,P_3,P_4,P_7,P_9\}} \right\}_{r=4} + \left\{ \frac{\{P_{10}\} \cup \{P_3,P_{10}\}}{\{P_3,P_{10}\}} \right\}_{r=7} + \left\{ \frac{\{P_8\} \cup \{P_8\}}{\{P_8\}} \right\}_{r=11} \end{array} \right\}$$

$$= \frac{2}{6} + \frac{1}{5}\left\{\frac{4}{5} + \frac{3}{7} + \frac{4}{6} + \frac{2}{2} + \frac{1}{1}\right\} = 1.11$$

4.2. Semantic relationship model

Only considering link structure on a complex network is inaccurate and not comprehensive. In this subsection, we define k-step index to compute the term weighting and focus on how to use k-step index to construct semantic relationship model. The attributes compose one or several large graphs representing the closeness degree between them. The objects can have their own term weighting representation through calculating the frequency of attributes and the closeness degree between them.

We take the bibliographic network as an example. An adjacent square matrix S, which consists of the same target objects with matrix L, is built to store the semantic relationship similarity between target objects. Thus,

$$S_{n\times n} = \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s_n} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ s_{n2} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

where s_{ij} represents the semantic relationship similarity between author a_i and author a_j. Then, we detail how to calculate s_{ij} using our proposed k-step index algorithm. Table 2 shows ten paper ID and the corresponding keywords represented by capitals A to L. The keywords constitutes the whole network $G' = (\mathcal{V}', \mathcal{E}')$ as shown in Fig. 2. The edge weight is the times that two keywords appear at the same paper. Suppose author a_p published papers P_1 and P_2, the term weighting of any keyword q that has direct correlation with a_p is calculated using Eq. (2).

$$w_{pq} = \frac{N(p) \cdot D(pq)}{K(p)} \cdot \log\left(\frac{N_s}{P(q)+1}\right) \tag{2}$$

where $N(p)$ is the number of papers published by author a_p. $D(pq)$ is the number of keyword q appearing in the papers published by a_p. Thus, the more a keyword appears in the papers published by a_p, the more significant it is in a_p's term weighting. $K(p)$ is the total keywords published by author a_p. N_s is the total number of papers in the network. $P(q)$ is the number of papers including keyword $P(q)$. That is, if a keyword appears at many papers in the collection frequently, it is not considered to be particularly representative of this kind of papers. For example, the keyword "data mining" is a broad research area. If an author publishes a paper with this keyword, we probably do not know what research area this paper is related to. Once the keyword becomes more specific, such as temporal outlier detection, outlier pair detection and so on, the feature can be more representative in term weighting. Therefore,

$$w_{p,C} = \frac{2\times 2}{6} \times \log\left(\frac{10}{7+1}\right) = 0.067$$

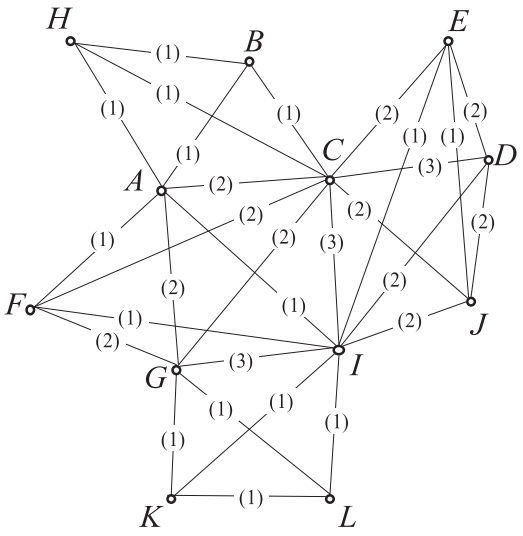


Fig. 2. The undirected weighted graph constructed by the keywords.

Only applying the attributes that have direct correlation with the target objects is not enough. The features, that have indirect correlation with the target objects, also need to be assigned with a term weighting. After obtaining the term weighting of keywords related to the authors directly, we perform k-step index to make the features be more complete. When k is zero, the subgraph of network G' is described in the left part of Fig. 3. The nodes, which are available for calculating term weighting, include A, B, C, D, E and H (known as starting nodes). Since graph may include circles, each vertex can be marked using an auxiliary array to prevent graph traversal algorithms from infinite loops. Then, each vertex, that is adjacent to the starting nodes, is visited under the condition that k is equal to 1. The nodes which are available for calculating term weighting, are the ones shown in the middle part of Fig. 3 except for starting nodes. Likewise, when k is equal to 2, the nodes K and L are available to be assigned with a term weighting. The term weighting of any keyword q that has indirect correlation with a_p is calculated using Eq. (3).

$$w_{pq} = \frac{1}{k+1} \cdot \frac{\sum_{u=1}^{t} \varphi_{uq} \cdot w_{pu}}{t} \tag{3}$$

where k is the step number from starting nodes to q. The node u is visited when the step number is $k-1$. φ_{uq} is edge weight between node u and node q. t is the number of node u. Thus, the term weighting calculation formula is summarized in Eq. (4) as follows. Then, each element s_{ij} in matrix S representing the similarity between a_i and a_j is calculated by cosine similarity.

$$w_{pq} = \begin{cases} \frac{N(p)\cdot D(pq)}{K(p)} \cdot \log\left(\frac{N_s}{P(q)+1}\right) & k = 0 \\ \frac{1}{k+1} \cdot \frac{\sum_{u=1}^{t} \varphi_{uq} \cdot w_{pu}}{t} & k > 0 \end{cases} \tag{4}$$

In addition, we set a threshold θ for index step k to prevent excessive time consumption. Once the parameter k reaches the threshold θ, the k-step index algorithm will terminate. We will detail how to choose θ in Section 6.3.

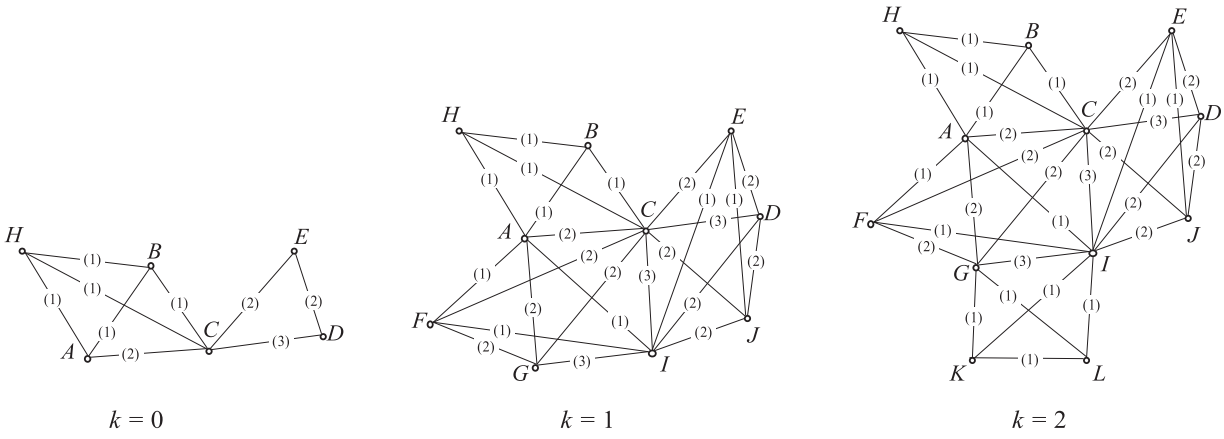


Fig. 3. The illustration of k-step index.

Algorithm of Calculating Term Weighting Using k-step Index

Input: (1) a weighted indirect graph G', (2) a target node a_p, (3) a threshold θ.
Output: The term weighting of target node a_p.
1. Initialize k to 0
2. Initialize array visited[graphsize]
3. **for** $v \leftarrow 0$ to graphsize
4. visited[v] $\leftarrow 0$
5. end for
6. Search for the nodes related to a_p directly and mark the nodes as starting nodes v'
7. **for each** starting node v'
8. calculating the term weighting $w_{pv'}$ of a_p using Eq. (2)
9. $w \leftarrow getNumber(v')$// get the number of node v'
10. visited[w] $\leftarrow 1$
11. end for
12. **while**$((getNeighbor(v') \neq \text{NULL})\&\&(k \leq \theta))$
13. $k \leftarrow k+1$
14. **for each** node v'' in $getNeighbor(v')$
15. $w \leftarrow getNumber(v'')$
16. **if**(visited[w]== 0)
17. calculating the term weighting $w_{pv''}$ of a_p using Eq. (3)
18. **end if**
19. **end for**
20. $v' \leftarrow v''$
21. end while

5. Detecting outlier pairs using LSModel

In this section, we will present our outlier pair detection method using LSModel for LSOutPair detection.

According to Definition 1, let vector $\boldsymbol{l_i}$ be x_1 and let $\boldsymbol{s_i}$ be x_2. Then, we perform linear transformation to each vector in matrices $L_{n\times n} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l_n}]^T$ and $S_{n\times n} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s_n}]^T$ and get the final similarity matrix $W_{n\times n}$ using Eq. (5) as follows.

$$W_{n\times n} = \begin{bmatrix} \alpha l_{11}+\beta s_{11} & \alpha l_{12}+\beta s_{12} & \cdots & \alpha l_{1n}+\beta s_{1n} \\ \alpha l_{21}+\beta s_{21} & \alpha l_{22}+\beta s_{22} & \cdots & \alpha l_{2n}+\beta s_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha l_{n1}+\beta s_{n1} & \alpha l_{n2}+\beta s_{n2} & \cdots & \alpha l_{nn}+\beta s_{nn} \end{bmatrix} = \alpha L_{n\times n} + \beta S_{n\times n} \quad (5)$$

subject *to* $\|\alpha L_{n\times n}\|_F - \|\beta S_{n\times n}\|_F < \varepsilon$,

where $\|\cdot\|_F$ is the Frobenius norm of a matrix: $\|\boldsymbol{A}\|_F = (\sum_{m,n} \boldsymbol{A}_{mn}^2)^{1/2}$.

At this time, the outlier pair matrix M_{os} can be calculated using the established α and β in Eq. (6) as follows. M_{os} is the absolute value of $\alpha\boldsymbol{L} - \beta\boldsymbol{S}$ whose (i, j) entry is $|l_{ij} - s_{ij}|$ for each ordered pair (i, j).

$$M_{os} = \begin{bmatrix} |\alpha l_{11}-\beta s_{11}| & |\alpha l_{12}-\beta s_{12}| & \cdots & |\alpha l_{1n}-\beta s_{1n}| \\ |\alpha l_{21}-\beta s_{21}| & |\alpha l_{22}-\beta s_{22}| & \cdots & |\alpha l_{2n}-\beta s_{2n}| \\ \vdots & \vdots & \vdots & \vdots \\ |\alpha l_{n1}-\beta s_{n1}| & |\alpha l_{n2}-\beta s_{n2}| & \cdots & |\alpha l_{nn}-\beta s_{nn}| \end{bmatrix} \quad (6)$$

Using the outlier pair matrix described above, we can obtain the maximum difference between link structure and semantic relationship of two target objects. The purpose of the condition $||\alpha L_{n\times n}||_F - ||\beta S_{n\times n}||_F < \varepsilon$ is to make sure that the link structure similarity and the semantic relationship similarity have the same order of magnitude. Once the values of $L(S)$ matrix are too high or too low, the matrix M_{os} will not get outlier pairs with good performance. Thus, we use parameters α and β to adjust that the Frobenius norm of matrix $\alpha L_{n\times n}$ and the Frobenius norm of matrix $\beta S_{n\times n}$ are almost the same, which can make the two matrices $\alpha L_{n\times n}$ and $\beta S_{n\times n}$ have the same order of magnitude. We first calculate the Frobenius norm of matrices L and S, respectivevly. If $||L||_F$ is greater than $||S||_F$, then α is set to 1. We increase β by 0.1 each time and iteratively compute $||\alpha L_{n\times n}||_F - ||\beta S_{n\times n}||_F$ to judge whether the difference is smaller than the threshold ε. On the contrary, we set β to 1 and increase α by 0.1 each time. The top-K values in M_{os} means the two target objects have the maximum difference between link structure and semantic relationship, which are regarded as outlier pairs in the whole complex network. At the same time, the matrix $W_{n\times n}$, as a byproduct in the outlier pair detection process, also can be applied to data analysis. The top-K values in $W_{n\times n}$ show that the link structure similarity and semantic relationship similarity are both very high, the target objects can be used to mine domain experts and identify authority Web pages, etc.

Algorithm of Outlier Pair Detection Using *LSModel*

Input: (1) a link structure matrix L, (2) a semantic relationship matrix S
Output: Top-K outlier pairs
1. Initialize α to 1
2. Initialize β to 1
3. Calculating the Frobenius norm of matrix L, $\|L\|_F$
4. Calculating the Frobenius norm of matrix S, $\|S\|_F$
5. **if** $\|L\|_F \geq \|S\|_F$ **then**
6. **while** $\|\alpha L_{n\times n}\|_F - \|\beta S_{n\times n}\|_F \geq \varepsilon$
7. $\beta \leftarrow \beta + 0.1$
8. **end while**
9. else
10. **while** $\|\alpha L_{n\times n}\|_F - \|\beta S_{n\times n}\|_F \geq \varepsilon$
11. $\alpha \leftarrow \alpha + 0.1$
12. **end while**
13. end if
14. Outlier pair matrix $M_{os} \leftarrow$ the absolute value of $\alpha\boldsymbol{L} - \beta\boldsymbol{S}$
15. The top-K values in M_{os} are regarded as the outlier pairs

6. Experiments and results

As we known, it is always a challenging problem of evaluating outliers due to the lack of ground truth. In this section, we define an outlierness measure, called *pOut*, for evaluating outlier pairs in complex network (Section 6.1). *precision* and *recall* are also used to be performance metrics (Section 6.1). Then, two datasets AMiner and Yahoo!Movies are used to validate the effectiveness of the proposed LSOutPair (Section 6.2). Several experiments are performed in the remainder of this paper to verify whether calculating the vast differences between link structure and semantic relationship holds for mining outlier pairs in complex networks (Section 6.3).

6.1. Performance metrics

In order to test the quality of our proposed outlier pair detection method, a novel performance metric is defined, which is called *pOut*. *pOut* can measure the number of object pairs that are marked wrongly or are missing. "Marked wrongly" means the object pair is a normal data but it is marked as an outlier pair. "Missing" means that the object pair should be an outlier pair but it does not exist in the top-K outlier pair set. The number of outlier pairs that are marked wrongly is denoted by $\mathcal{W}$. The number of outlier pairs that are missing during the outlier detection process is denoted by $\mathcal{M}$. *outPair* is the total outlier pairs that are annotated manually in the dataset. Accordingly, we define *pOut* using Eq. (7) as follows.

$$pOut = \frac{\mathcal{W} + \mathcal{M}}{2 \times |outPair|} \times 100\% \tag{7}$$

Contrary to another popular evaluation metric, called accuracy, *pOut* uses two opposite situations: true negative and false positive, to test whether all the possible outlier pairs are identified during the process of detection. Another two common metrics, *precision* and *recall*, are also applied to reflect the availability of our proposed detection model (Liu, 2011). *precision* for outlier pair detection is the fraction of object pairs assigned that are identified as top-K outlier pairs in the outlier pair matrix, which measures how well it is doing at rejecting normal object pairs. *recall* is the fraction of object pairs assigned by the manual annotation data, which measures how well it is doing at finding all the outlier pairs. Therefore, *precision* and *recall* are calculated using Eqs. (8) and (9) as follows.

$$precision = \frac{|pairD| - \mathcal{W}}{|pairD|} \times 100\% \tag{8}$$

$$recall = \frac{|pairD| - \mathcal{W}}{|outPair|} \times 100\% \tag{9}$$

where *pairD* is the set of object pairs in the top-K outlier pairs of the outlier pair matrix. *outPair* is the total outlier pairs that are annotated manually in the dataset. *F-Measure* (Croft, Metzler, & Strohman, 2009), as the harmonic mean of *precision* and *recall*, is also used to measure the performance of our method. It is calculated using Eq. (10) as follows.

$$F - Measure = \frac{(\gamma + 1)\, precision \times recall}{\gamma^2 \times precision + recall} \tag{10}$$

where γ is a weight for reflecting the relative importance of *precision* and *recall*. Obviously, if $\gamma > 1$, then the *recall* value is more important than the *precision* value. In this paper, γ is assigned a constant 1.

6.2. Datasets

We perform the experiments using two real datasets: AMiner (Tang et al., 2008) and Yahoo!Movies (Yahoo! webscope program, 2016).

We generate data from AMiner, which is a bibliographic information network. It has three main parts including AMiner-Author.txt, AMiner-Paper.txt, and AMiner-Coauthor.txt. It has 1,712,433 authors and 2,092,356 papers covering different areas in computer science. We choose 10,000 users and the corresponding information in the experiments. There are four types of nodes: paper, author, venue, and term and several types of edges: writing and written-by, publishing and published-by, using and used-by, constructing the whole information network. In order to detect outliers with higher accuracy, a supplement is added to the original dataset. The keywords of each paper are extracted using our crawler (denoted by #k) (Peng & Liu 2013), and they are added to the end of each record in AMiner-Paper.txt. There are one hundred outlier pairs data annotated manually in each dataset. The formats of AMiner-Author.txt and AMiner.Paper.txt are described as follows:

AMiner-Author.txt:
#index —- index id of this author
#n —- name (separated by semicolons)
#a —- affiliations (separated by semicolons)
#pc —- the count of published papers of this author
#cn —- the total number of citations of this author
#hi —- the H-index of this author
#pi —- the P-index with equal A-index of this author
#upi —- the P-index with unequal A-index of this author
#t —- extracted key terms of this author (separated by semicolons)

AMiner.Paper.txt:
#index —- index id of this paper
#*—- the title of this paper
#@ —- the author names of this paper
#o —- affiliations (separated by semicolons)
#t —- year of publication
#c —- publication name

Yahoo!Movies, as a part of rating and classification dataset, can be applied to in complex information network. This dataset includes six files: ydata-ymovies-user-movie-ratings-train-v1_0.txt, ydata-ymovies-user-movie-ratings-test-v1_0.txt, ydata-ymovies-user-demographics-v1_0.txt, ydata-ymovies-movie-content-descr-v1_0.txt, ydata-ymovies-mapping-to-mov ielens-v1_0.txt, ydata-ymovies-mapping-to-eachmovie-v1_0.txt, covering various information about movies, actors, movie rating and so on. The multi-typed vertices and the multi-typed relations between them can be used for classification, clustering or detecting outliers. The rating information can be used for predicting or recommendation system. For example, the format of "ydata-ymovies-movie-content-descr-v1_0.txt" is described as follows:

1 title
2 synopsis
3 running time
4 MPAA rating
5 reasons for the MPAA rating
.....

We use three types of objects: actor, movie, genres and their relations: actor–starring–movie, movie–belonging to–genres, to construct the movie network. The whole dataset contains 7642 actors, but we only choose 5000 actors and their movie information in the experiments. Also, one hundred object pairs are added in the dataset as outlier pairs.

6.3. Results

In this section, we conduct five experiments to examine the effectiveness and efficiency of our proposed outlier pair detection method. The first experiment is conducted to find the most suitable α and β when obtaining the outlier pair matrix. We carry on

Table 3
The *pOut, precision, recall* and *F-Measure* of LSOutPair on AMiner and Yahoo!Movies. ($k = 10$, Top-100 object pairs are selected as outlier pairs).

α	β	AMiner				α	β	Yahoo!Movies			
		pOut	*Precision*	*Recall*	*F-Measure*			*pOut*	*Precision*	*Recall*	*F-Measure*
1	2.6	0.65	0.35	0.35	0.35	3.3	1	0.76	0.28	0.28	0.28
1	2.7	0.58	0.42	0.42	0.42	3.4	1	0.6	0.4	0.4	0.4
1	2.8	0.36	0.64	0.64	0.64	3.5	1	0.39	0.61	0.61	0.61
1	2.9	0.21	0.79	0.79	0.79	3.6	1	0.24	0.76	0.76	0.76
1	3.0	0.12	0.88	0.88	0.88	3.7	1	0.16	0.86	0.86	0.86
1	**3.1**	**0.07**	**0.93**	**0.93**	**0.93**	**3.8**	**1**	**0.1**	**0.9**	**0.9**	**0.9**
1	3.2	0.11	0.89	0.89	0.89	3.9	1	0.16	0.84	0.84	0.84
1	3.3	0.24	0.76	0.76	0.76	4.0	1	0.27	0.73	0.73	0.73
1	3.4	0.38	0.62	0.62	0.62	4.1	1	0.41	0.59	0.59	0.59
1	3.5	0.50	0.50	0.50	0.50	4.2	1	0.62	0.38	0.38	0.38

Table 4
The *pOut, precision, recall* and *F-Measure* of LSOutPair on AMiner and Yahoo!Movies. ($k = 10$, Top-50 object pairs are selected as outlier pairs).

α	β	AMiner				α	β	Yahoo!Movies			
		pOut	*Precision*	*Recall*	*F-Measure*			*pOut*	*Precision*	*Recall*	*F-Measure*
1	2.6	0.46	0.58	0.29	0.387	3.1	1	0.42	0.66	0.33	0.44
1	2.7	0.4	0.7	0.35	0.467	3.2	1	0.36	0.78	0.39	0.52
1	2.8	0.33	0.84	0.42	0.56	3.3	1	0.31	0.88	0.44	0.587
1	**2.9**	**0.26**	**0.98**	**0.49**	**0.653**	**3.4**	**1**	**0.28**	**0.94**	**0.47**	**0.627**
1	3.0	0.32	0.86	0.43	0.573	3.5	1	0.3	0.9	0.45	0.6
1	3.1	0.36	0.78	0.39	0.52	3.6	1	0.33	0.84	0.42	0.56
1	3.2	0.41	0.68	0.34	0.453	3.7	1	0.37	0.76	0.38	0.507
1	3.3	0.45	0.6	0.3	0.4	3.8	1	0.41	0.68	0.34	0.453
1	3.4	0.49	0.52	0.26	0.347	3.9	1	0.45	0.6	0.3	0.4
1	3.5	0.57	0.36	0.18	0.24	4.0	1	0.48	0.54	0.27	0.36

a large number of experiments with various parameter settings. When the Frobenius norm of matrix L is greater than the Frobenius norm of matrix S, we vary the parameter β. Otherwise, we vary the parameter α. Tables 3 and 4 show the results on two datasets and the corresponding parameter settings. Note that since we choose top-100 values and the corresponding object pairs as outlier pairs in the Table 3, $|pairD|$ is the equal to $|outPair|$ in this case, which makes the *precision* is the same as *recall*. In addition, the sum of *pOut* and *precision* is 1. We choose top-50 values in outlier pair matrix and the corresponding object pairs as outlier pairs in Table 4. Once the number of detected outlier pairs is changed, we can vary α and β to make *F-Measure* reach its peak. In Table 3, the results show that α and β should be set to 1 and 3.1 respectively in AMiner, and α and β should be set to 3.8 and 1 in Yahoo!Movies. It can be seen from the results that when the number of attributes of the object is considerable, α is usually less than β. For example, in bibliographic network, there are many term attributes of an author after k-step index. Then, α is less than β in AMiner. On the contrary, when the number of attributes of the object is relatively small, α is usually greater than β. In addition, the results of top-50 outlier pairs are listed in Table 4, the *precision* becomes higher than that in Table 3. However, *recall* decreases a lot because there are one hundred outlier pairs annotated manually beforehand, that is, $|outPair| = 2|pairD|$.

In the second experiment, we attempt to find the most appropriate θ during the process of k-step index. As described in Section 4.2, the greater the parameter k is, the more accurate the term weighting of an object is. However, taking one more index step will increase the time consumption. Then, we vary the parameter θ from 1 to 10. As we can see from Fig. 4, *F-Measure* almost stays unchanged when θ reaches 4 in AMiner. Similarly, the parameter θ should be set to 7 in Yahoo!Movies. Because the number of attributes in AMiner is much more than that in Yahoo!Movies,

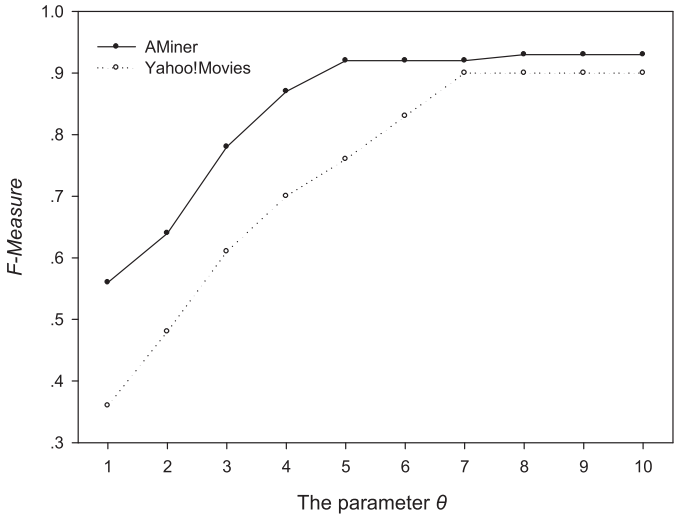


Fig. 4. Dynamic plot of average *F-Measure* versus parameter θ on AMiner and Yahoo!Movies. (Top-100 object pairs are selected as outlier pairs).

it needs a greater θ during the process of k-step index in Yahoo!Movies to get a more accurate term weighting of objects.

In the third experiment, we study the impact of threshold ε on the accuracy of results. The parameter ε is increased from 1000 to 10,000 in AMiner dataset (and ε is increased from 400 to 5000 in Yahoo!Movies dataset) to observe the *F-Measure* values of outlier pair detection. When the parameter ε is small, it means that the parameter ε has a high requirement on the difference between link structure similarity and semantic relationship similarity. However, the difference between link structure similarity and semantic relationship similarity, that is, the difference between the Frobenius norm L matrix and S matrix, may never be less than ε. This

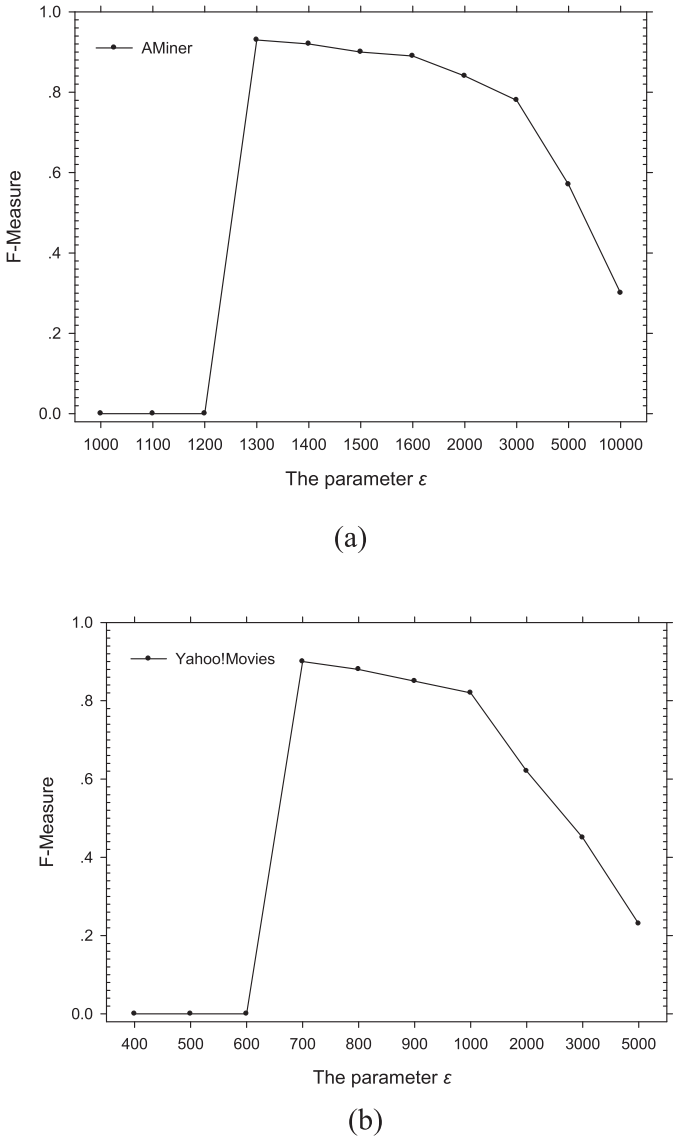


Fig. 5. Dynamic plot of average *F-Measure* versus parameter ε on AMiner and Yahoo!Movies.

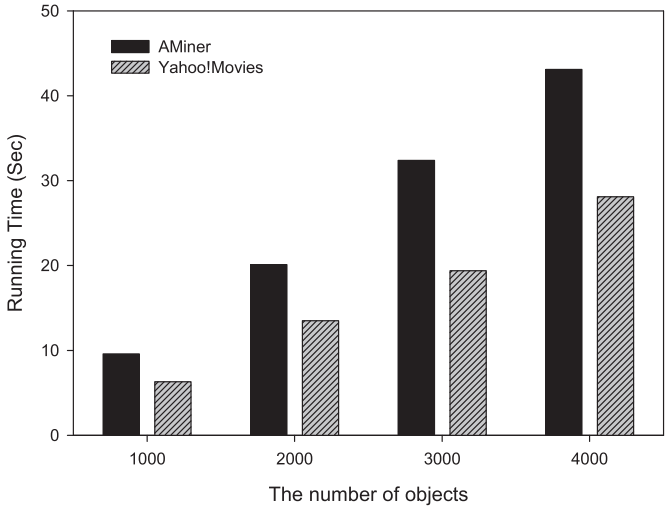


Fig. 6. The running time comparison on the condition of different number of data over AMiner and Yahoo!Movies datasets. (Top-100 object pairs are selected as outlier pairs).

is why the value of *F-Measure* is zero when ε is equal or smaller than 1200. A high parameter ε may also cause the situation that two matrices do not have the same order of magnitude, which will impact the performance of our proposed method. According to the curves in Fig. 5, we set ε to 1300 and 700 in AMiner and Yahoo!Movies, respectively. Since different datasets have different size, it is almost impossible to make a fixed parameter epsilon (or theta) suitable for all datasets. But once we get the appropriate values of the thresholds, we do not need to run the proposed approach again when testing the new data.

In real situation, the Frobenius norms of link structure similarity and semantic relationship similarity are calculated first. According to the smaller value, we can estimate appropriate ε and θ to observe how α and β influence the final ranking. When α and β are fixed, we can optimize ε and θ.

In the fourth experiment, we verify the scalability of our proposed algorithm. In AMiner and Yahoo!Movies datasets, we increase the number of objects from 1000 to 4000, and then observe the running time. Fig. 6 shows that the execution time is almost linear growth instead of exponential growth with the linear growth of data volume.

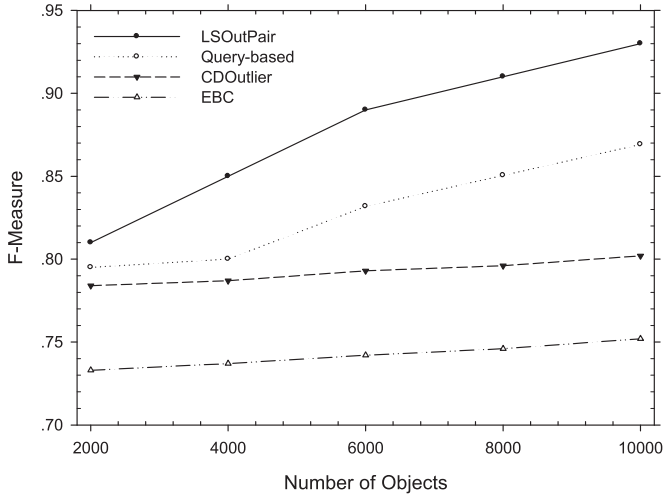
The fifth experiment is conducted to compare the performance of our proposed method and three baseline algorithms (CDOutlier (Gupta et al., 2013a), EBC (Gupta, Gao, Yan, Cam, & Han, 2013b) and Query-based outlier detection (Kuck, Zhuang, Yan, Cam, & Han, 2015)). CDOutlier discovers popular community distribution patterns for all the object types based on joint non-negative matrix factorization. CDOutlier groups authors based their research area distributions. That is, it only considers the semantic information in the network. EBC groups the attributes of an object individually. Whether the object is marked as an outlier depends on the number of anomalous attributes. The number of the attributes in a dataset may influence the final performance of outlier detection. Query-based outlier detection finds anomalies according to the queries input by users. It considers more link information than semantic information in the whole process. According to the curves in Fig. 7, LSOutPair performs better than Query-based outlier detection, CDOutlier and EBC. The *F-Measure* increases slowly with the increasing number of objects. But, in LSOutPair, more objects are involved in the experiments means that the link similarity and semantic similarity should be more accurate. Table 5 provides a summary of LSOutPair and three baseline algorithms.

7. Discussion

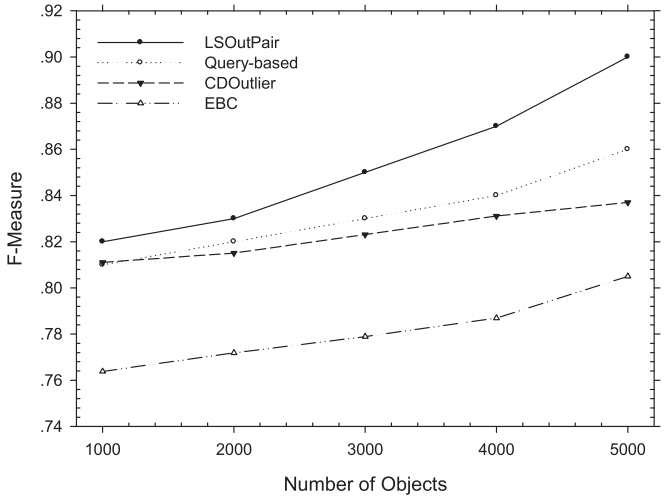
Most traditional outlier detection methods are used to identify single outlier or outlier subgraph in complex network or discover outliers in homogenous network. The outlier pairs in homogeneous network can be obtained only based on the statistic information. A lot of link information or semantic information is ignored. As described in Table 5, LSOutPair is an outlier pair detection method in complex network. The main advantages of LSOutPair are: (1) it considers the link structure and semantic relationship of the objects in the datasets, (2) using the difference between link structure similarity and semantic relationship similarity to detect outlier pair is a new attempt in data mining, (3) it combines multi-typed objects and multi-typed relations instead of only using single type objects. However, LSOutPair also has its limitations: when calculating link structure similarity, we just use two types of objects “author” and “paper”, and their link relations. At this time, the terms are ignored. When calculating semantic relationship similarity, we also use three types of objects “author”, “paper”, and “term”, and their link relations. Although we only deal with “term”, the “paper” information plays an important role in

Table 5
A summary of LSOutPair and three baseline algorithms.

Methods	Multi-typed network	Type of outliers	Link-based	Semantic-based	Difference between link and semantics	User interaction
LSOutPair	Yes	Outlier Pair	Yes	Yes	Yes	No
Query-based	Yes	Outlier	Yes	No	No	Yes
CDOutlier	Yes	Outlier	No	Yes	No	No
EBC	Yes	Outlier	No	Yes	No	No



(a) AMiner



(b) Yahoo!Movies

Fig. 7. Performance comparison of four outlier (pair) detection methods on AMiner and Yahoo!Movies.

connecting "author" and "term". In link similarity part, two types "writing" and "writing-by" are used. In semantic similarity part, four types "writing" and "writing-by", "using" and "used-by" are used. Only utilizing "term" in the process of calculating semantic similarity may be a disadvantage of this paper. But multi-typed objects and multi-typed links are considered in the link model and semantic model. In future research, we may consider combines multi-typed objects in the process of feature representation. In addition, the users need to tune the four parameters to get the final results when training the data in a new dataset, which is another limitation of this method.

8. Conclusions

In this paper, we propose an outlier pair detection method for complex information networks based on link structure and semantic relationship. We define an LSModel, which includes *Link* part reflecting the link structure similarity between objects and *Semantic* part representing the semantic similarity between objects on the network. A *k*-step index algorithm is introduced to compute the term weighting of objects to obtain the semantic similarity. The linear transformation and Frobenius norm are combined to find differences between *Link* part and *Semantic* part and rank the outlierness for outlier pair. We also propose *pOut*, as a measurement for outlier pair, to evaluate effectiveness of the proposed LSOutPair. Experimental results show that our proposed outlier detection method, LSOutPair, can discover outlier pairs in c networks effectively and efficiently.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant No. 60903098, the Project of Jilin Provincial Industrial Technology Research and Development (JF2012c016-2), and Graduate Innovation Fund of Jilin University (2016183, 2016184).

References

Aggarwal, C. C., & Sathe, S. (2015). Theoretical foundations and algorithms for outlier ensembles. *ACM SIGKDD Explorations Newsletter, 17*(1), 24–47.
Akoglu, L., Tong, H., & Koutra, D. (2014). Graph-based anomaly detection and description: A survey. *Data Mining & Knowledge Discovery, 29*(3), 626–688.
Angiulli, F., Basta, S., Lodi, S., & Sartori, C. (2013). Distributed strategies for mining outliers in large data sets. *IEEE Transactions on Knowledge and Data Engineering, 25*(7), 1520–1532.
Angiulli, F., & Fassetti, F. (2016). Towards generalizing the unification with statistical outliers: The gradient outlier factor measure. *ACM Transactions on Knowledge Discovery from Data (TKDD), 10*(3) Article 27.
Aydin, I., Karakose, M., & Akin, E. (2015). Anomaly detection using a modified kernel-based in the pantograph-catenary system. *Expert Systems with Applications, 42*(2), 938–948.
Cao, L., Wei, M. R., Yang, D., & Rundensteiner, E. A. (2015). Online outlier exploration over large datasets. In *KDD '15 proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 89–98).
Croft, W. B., Metzler, D., & Strohman, T. (2009). *Search engines: Information retrieval in practice*. Boston: Addison-Wesley.
Dai, H. B., Zhu, F. D., Lim, E. P., & Pang, H. (2015). Detecting anomaly collections using extreme feature ranks. *Data Mining and Knowledge Discovery, 29*(3), 689–731.
Dalmia, A., Gupta, M., & Varma, V. (2015). Query-based graph cuboid outlier detection. In *Proceedings of the 2015 IEEE/ACM international conference on advances in social network analysis and mining (ASONAM 2015)*.
Dufrenois, F., & Noyer, J. C. (2016). One class proximal support vector machines. *Pattern Recognition, 52*, 96–112.
Fusters, D., Dafonte, C., Arcay, B., Manteiga, M., Smith, K., Vallenari, A., et al. (2013). SOM ensemble for supervised outlier analysis. Application to outlier identification in the Gaia astronomical survey. *Expert Systems with Applications, 40*(5), 1530–1541.
Gupta, M., Gao, J., Aggarwal, C., & Han, J. (2013a). Community distribution outlier detection in heterogeneous information networks. In *European conference on machine learning and principles and practice of knowledge discovery in databases (ECML-PKDD 2013)* (pp. 557–573).
Gupta, M., Gao, J., Sun, Y., & Han, J. (2012a). Community trend outlier detection using soft temporal pattern mining. In *Proceedings of 2012 european conference on machine learning and principles and practice of knowledge discovery in databases (ECMLPKDD'12)*. Bristol, UK.
Gupta, M., Gao, J., Sun, Y., & Han, J. (2012b). Integrating community matching and outlier detection for mining evolutionary community outliers. In *Proceedings of 2012 ACM SIGKDD conference on knowledge discovery and data mining (KDD'12)*.

Gupta, M., Gao, J., Yan, X. F., Cam, H., & Han, J. (2013b). On detecting association-based clique outliers in heterogeneous information networks. In Advances in social networks analysis and mining, ASONAM'13 (pp. 108–115).
Gupta, M., Mallya, A., Roy, S., Cho, J. H. D., & Han, J. (2014). Local learning for mining outlier subgraphs from network datasets. In *Proceedings of the 2014 SIAM international conference on data mining*.
Hawkins, D. M. (1980). Identification of outliers. *Monograph on Applied Probability and Statistics*. Berlin: Springer.
Karppa, M., Kaski, P., & Kohonen, J. (2016). A faster subquadratic algorithm for finding outlier correlations. In *SODA '16 Proceedings of the twenty-seventh annual ACM-SIAM symposium on discrete algorithms* (pp. 1288–1305).
Kim, S., Cho, N. W., Lee, Y. J., Kang, S. H., & Kim, T. (2013). Application of density-based outlier detection to database activity monitoring. *Information Systems Frontiers, 15*(1), 55–65.
Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications, 41*(4), 1690–1700.
Kuck, J., Zhuang, H. L., Yan, X. F., Cam, H., & Han, J. W. (2015). Query-based outlier detection in heterogeneous information networks. In *Proceedings of the 18th international conference on extending database technology* (pp. 325–336).
Lei, P. R. (2016). A framework for anomaly detection in maritime trajectory behavior. *Knowledge & Information Systems, 47*(1), 189–214.
Leon, S. J. (2011). *Linear algebra with applications* (eighth edition). China, Beijing: Machine Press.
Liu, B. (2011). *Web data mining: Exploring hyperlinks, contents and usage data* (2nd ed.). Berlin: Springer.
Maervoet, J., Vens, C., Berghe, G. V., Blockeel, H., & DeCausmaecker, P. (2012). Outlier detection in relational data: A case study in geographical information systems. *Expert Systems with Applications, 39*(5), 4718–4728.
Peng, T., & Liu, L. (2013). Focused crawling enhanced by CBP–SLC. *Knowledge-based Systems, 51*, 15–26.
Rossi, R. A., Gallagher, B., Neville, J., & Henderson, K. (2013). Modeling dynamic behavior in large evolving graph. In *Proceeding of the 6th ACM international conference on web search and data mining (WSDM)* (pp. 667–676).
Tang, J., Zhang, J., Yao, L. M., Li, J. Z., Zhang, L., & Su, Z. (2008). Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 990–998).
Xie, S. H., Wang, G., Lin, S. Y., & Yu, P. S. (2007). Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD)* (pp. 824–833).
Yahoo! webscope program. http://webscope.sandbox.yahoo.com Accessed: 28/01/2016.
Yao, Z., Mark, P., & Rabbat, M. (2012). Anomaly detection using proximity graph and PageRank algorithm. *IEEE Transactions on Information Forensics and Security, 7*(4), 1288–1300.
Zhuang, H., Zhang, J., Brova, G., Tang, J., Cam, H., Yan, X., et al. (2014). Mining query-based subnetwork outliers in heterogeneous information networks. In *IEEE international conference on data mining* (pp. 1127–1132).
Zhang, Y., Lu, H. C., Zhang, L. H., & Ruan, X. (2016). Combining motion and appearance cues for anomaly detection. *Pattern Recognition, 51*, 443–452.