Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 125123, 7 pages
http://dx.doi.org/10.1155/2013/125123

Research Article

The Algorithm of Link Prediction on Social Network

Liyan Dong,[1,2] Yongli Li,[3] Han Yin,[1,2] Huang Le,[1,2] and Mao Rui[1,2]

[1] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2] *Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China*
[3] *School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China*

Correspondence should be addressed to Yongli Li; liyl603@nenu.edu.cn

Received 20 May 2013; Revised 17 August 2013; Accepted 17 August 2013

Academic Editor: William Guo

At present, most link prediction algorithms are based on the similarity between two entities. Social network topology information is one of the main sources to design the similarity function between entities. But the existing link prediction algorithms do not apply the network topology information sufficiently. For lack of traditional link prediction algorithms, we propose two improved algorithms: CNGF algorithm based on local information and KatzGF algorithm based on global information network. For the defect of the stationary of social network, we also provide the link prediction algorithm based on nodes multiple attributes information. Finally, we verified these algorithms on DBLP data set, and the experimental results show that the performance of the improved algorithm is superior to that of the traditional link prediction algorithm.

1. Introduction

Currently with the rapid development, online social network has been a part of people's life. A lot of sociology, biology, and information systems can use the network to describe, in which nodes represent individual and edges represent the relationships between individuals or the interaction between individuals. Therefore, the study of complex networks has been the important branch of many scientific fields. Link prediction is an important task in link mining. Link prediction is to predict whether there will be links between two nodes based on the attribute information and the observed existing link information. Link prediction not only can be used in the field of social network but can also be applied in other fields. As in bioinformatics, link prediction can be used to discover interactions between proteins [1]; in the field of electronic commerce, link prediction can be used to create the recommendation system [2]; and in the security field, link prediction can help to find the hidden terrorist criminal gangs [3]. Link prediction is closely related to many areas. Therefore, in recent years there are a lot of correlation algorithms proposed to solve the problem of link prediction.

2. The Summary of Social Network

In real life, the individuals are not independent of each other. They are mutually contacted and affected. If we only pay attention to individual attributes, while ignoring the relationships between individuals, this is bound to affect the accuracy and comprehensiveness of analysis. Social network represents the relationship between social entities (such as each person, social group). Social network analysis focuses on explaining the hidden patterns and the effects of these relationships. It is based on such an assumption, namely, the individuals in social groups are interdependent, not independent autonomous units. Social network includes a set of objects and relationships between them [4]. These relationships can be any type of social relationships, such as friendship, the purchase of relationship. Social networks can be represented by graph. Graph G contains nodes set V and edges set E. We can use $G = \langle V, E \rangle$ to represent the graph G. In this paper, the nodes set represents objects, and the edges set represents the relationships between objects.

The properties of social network are small world effect, scale-free effect, and cluster effect. Small world effect is

produced by American psychologist Stanley Milgram in 1969 [5]. He found that assign the name of the recipient randomly, and send a message to his own people, and so on, this message can reach the hands of the recipient in a relatively short path (about 6 people). The result was that the famous "six degrees of separation" theory was generated. Social network, internet network, and simulation network have small world properties. In the network, small world effect refers to that the average distance in the network is very small compared to the size of the network. That is to say, each pair of nodes can be connected through a short path in a network [6]. The scale-free effect refers to that most nodes' links are very small in the network; only a few nodes have lots of links. In this network, nodes with high degree are called hubs (hinge node). The hub node dominates the network operation. Scale-free effect displays that node degree distribution is seriously uneven in large-scale network. Clustering effect of social network refers to that there is a circle of friends, acquaintances, rings, and other small groups in social network. Each member of the small group knows each other. This phenomenon can be described by graph; namely, there is many fully connected subgraphs in social network.

3. The Traditional Link Prediction Algorithms

The link prediction is an important research field in data mining. It has a wide range of scenarios. Many data mining tasks involve the relationship between the objects. Link prediction can be used for recommendation systems, social networks, information retrieval, and many other fields.

Given a snapshot graph of the social network at a moment $G = \langle V, E\rangle$ and the node v_i and the node v_j, link prediction is to predict the probability of the link between the node v_i and the node v_j. It can be seen through the definition of link prediction that the link prediction task is divided into two categories. The first category is to predict that the new link will appear in future time. The second category is to forecast hidden unknown link in the space.

The easiest framework of link prediction algorithm is based on the similarity of the algorithm. Any pair of node x and node y, we have assigned to this node is a function *Similarity* (x, y), this function is defined as the similarity function between nodes x and y. Then sorting the nodes pair in accordance with the function values from the largest to smallest, the greater the value of the similarity function, the greater the probability of the link in the nodes.

Here we introduce some simple link prediction similarity indexes.

3.1. Local Similarity Index

3.1.1. Common Neighbors. Assume that the node $v \in V$; then the neighbors of the node set $\Gamma(v) := \{t \mid (t, v) \in E \vee (v, t) \in E \wedge t \neq v\}$; that is, $\Gamma(v)$ is the set of all the neighbors of node v. The common neighbors of node u and node v refer to the jointly owned neighbors by node u and node v. For the undirected graph, the common neighbors can use the following definition:

$$\text{Similarity}(u, v) = |\Gamma(u) \cap \Gamma(v)|. \tag{1}$$

Kossinets and Watts analysis of the large-scale social network, found that two students who have more mutual friends will have greater possibility to become friends [7].

3.1.2. Preferential Attachment. Preferential attachment mechanism can be used to generate scale-free network evolution model. The probability of generating a new link of node u is directly proportional to the degree of the node [8]. This is the same as the truth "the rich are getting richer" in economics. Therefore, the probability of the link between node u and node v is directly proportional to $d_u \times d_v$. Inspired by this mechanism, the PA similarity index can be defined as follows:

$$\text{Similarity}(u, v) = d_u \times d_v. \tag{2}$$

It may be noted that the similarity index does not require any node neighbor information; therefore, this similarity index has the lowest computational complexity.

3.1.3. Adamic-Adar [9]. This similarity index assigns a higher similarity function value to a small degree node. Adamic-Adar algorithm believes that an affair owned by less objects, compared to owned by more objects, has greater effect on link prediction. Its definition is as follows:

$$\text{Similarity}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log d_z}. \tag{3}$$

3.1.4. Resource Allocation. This similarity index is inspired by the ideas of complex network resources dynamically allocated [10]. In pair of nodes u, v that have no direct link, node u can allocate some resources to the node v through their common neighbor. Their common neighbors assume the role of passers. In the simplest case, we assume that each passer has a unit of resources; it assigns these resources to its neighbors evenly. Therefore, the similarity of node u and node v can be defined as the number of resources that node u get from node v; namely,

$$\text{Similarity}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{d_z}. \tag{4}$$

3.2. Overall Similarity Index

3.2.1. Katz [11]. In 1953, Katz described the similarity using the global path. The idea of the method is that the more paths between two nodes are, the greater the similarity between two nodes is. Katz measure is defined as follows:

$$\begin{aligned}\text{Similarity}(u, v) &= \sum_{l=1}^{l_{\max}=\infty} \beta^l \cdot \left|\text{path}_{u,v}^l\right| \\ &= \beta A_{uv} + \beta^2 \left(A^2\right)_{uv} + \beta^3 \left(A^3\right)_{uv} + \cdots,\end{aligned} \tag{5}$$

where $|\text{path}_{u,v}^{l}|$ is the number of paths between node u and node v and the length of the path is l. β is a parameter between 0 and 1. This parameter is used to control the contribution of path to the similarity; the longer the path is, the less contribution the path made to the similarity. The computational complexity of Katz measure is n^3, so the measure is not suitable for large-scale network.

3.2.2. Random Walk with Restart (RWR) [12]. This indicator is a direct application of the PageRank algorithm. A random walker starting from node u will reach its random neighbor with probability c repeatedly and return the node u with the probability $1 - c$. q_{uv} represents the probability of the random walker reaching node v in the steady state condition. Therefore, we have $\vec{q}_u = cP^T\vec{q}_u + (1 - c)\vec{e}_u$, where P is the transfer matrix. If the node u is connected with node v, then $P_{uv} = 1/d(u)$; else $P_{uv} = 0$. So the solution is simple; namely, $\vec{q}_u = (1 - c)(I - cP^T)^{-1}\vec{e}_u$. RWR coefficient can be defined as

$$\text{Similarity}(u, v) = q_{uv} + q_{vu}. \tag{6}$$

Compared to the local similarity index, the global similarity index needs more overall network topology information. Although the performance of the overall similarity index is better than the local similarity index, it has two fatal flaws: first, the global similarity index calculation is very time consuming, and when the network is huge, this calculation program of the global similarity index does not work. Second, sometimes the global topology information is not available, especially when we use a decentralized approach to implement the algorithm. Therefore, how to design a similarity index is particularly important, which is easy to calculate and its accuracy is high.

Although the traditional link prediction algorithms have made some prediction effect, they do not make full use of the topology information. Common neighbor algorithm treats all the common neighbors equally; it does not distinguish the different neighbors' different effects on the link prediction. Katz algorithm distinguishes the different path's influences which have different lengths, but it does not distinguish the influence of the paths with the same length on link prediction. These algorithms only consider the topology characteristics of the network, treat the social networking static, and ignore the time attributes and node attribute of social network. How to integrate the topology characteristics, time characteristics, and node attributes of social network reasonably is an enormous challenge for link prediction facing.

4. Link Prediction Algorithms Based on Node Guidance Capability

If there is the case in Figure 1, the traditional link prediction algorithm think the degree of similarity calculated between node A and node B is the same as the similarity between node X and node Y. However, when we extracted the subgraph that contained the node A, node B, and their common neighbors, the node X, node Y, and their common neighbors, as shown in Figure 2, we can see that the paths between node A and node B are more than the paths between the node X and the node Y. According to Katz algorithm the similarity of node A and node B is higher than the similarity of node X and node Y.

Observing the density of the extracted common neighbor subgraph, if the common neighbors subgraph is denser, the nodes in the subgraph made more contribution to link prediction. Assigning the dense of the subgraph to each node, we can see that if the common neighbor occupied the greater the proportion in the neighbor of the node, it has greater ability to generate new link between the node A and node B. Therefore, we designed the following formula to measure the guiding force of the node:

$$\text{Guidance} - \text{force}(z) = \frac{|\phi(z)|}{\log d_z}, \tag{7}$$

where d_z is the degree of node z and $|\phi(z)|$ is the degree of node z in the extracted subgraph.

4.1. CNGF Algorithm. By introducing the definition of node guidance capability, we know that the guidance capabilities of the different nodes are different. The similarity between the two nodes can be represented by the sum of each node's guidance capability. The guidance capability of the common neighbors is greater and the likelihood of the new link between the two nodes is greater. On this basis, we define the formula of the similarity degree of two nodes:

$$\text{Similarity}^{\text{CNGF}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{|\phi(z)|}{\log d_z}, \tag{8}$$

where $|\phi(z)|$ is the number of links that the node connected with the common neighbors.

With the previous formula, we can give the pseudocode of CNGF algorithm based on node guidance capability (see Algorithm 1).

The computational complexity of CNGF algorithm is $O(N^2)$; N is the maximum of all the nodes' degree. This computational complexity is also acceptable in large-scale network.

4.2. KatzGF Algorithm. The idea of Katz algorithm is that if there are more paths between two nodes, the possibility of a new link existing in the two nodes is greater. Katz algorithm did not distinguish the contribution of the paths with the same path length. The node guidance capability proposed in this paper can distinguish the contribution of different nodes effectively. We can make use of the node guidance capability in Katz algorithm. Because of different nodes, the contribution of the paths with the same path length is different. Based on the previous ideas, we designed the KatzGF algorithm. The KatzGF algorithms integrated the local node information and the global social network information reasonably.

The most important part of the KatzGF algorithm is how to design the formula for the degree of similarity between two nodes. Taking into account the contribution of the different

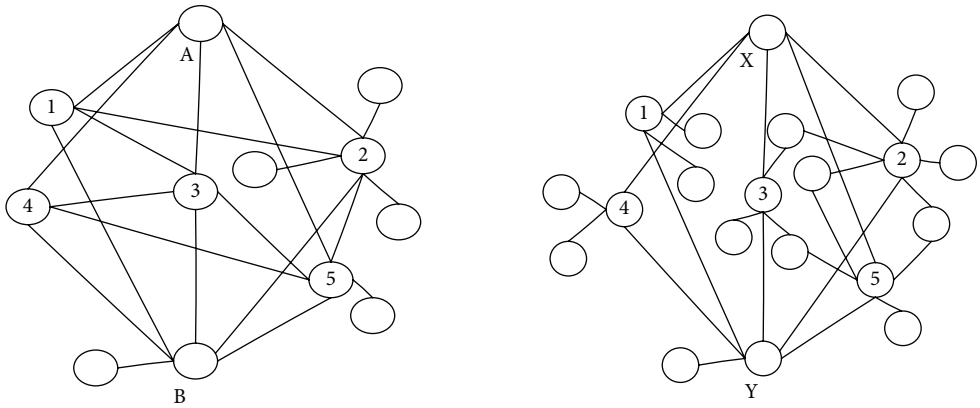

Figure 1: Two social network graphs with the same node degree.

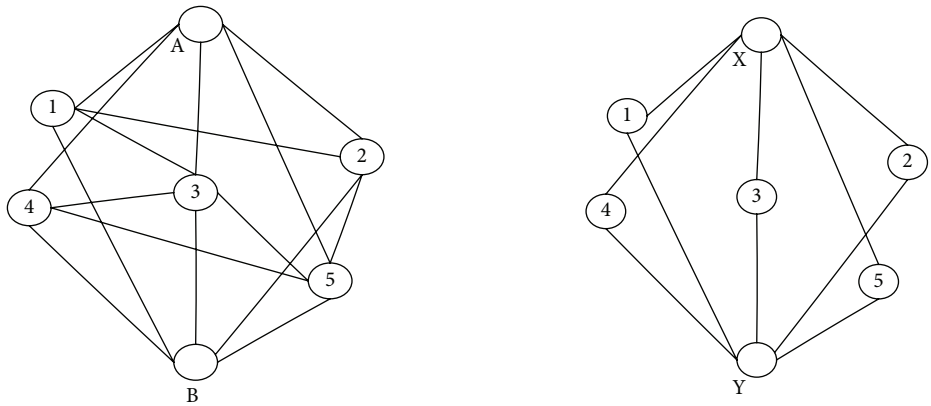

Figure 2: The extracted subgraph contains the prediction nodes and their common neighbor.

nodes on the same path length is different, we can design the similarity formula of KatzGF:

$$\text{Similarity}^{\text{KatzGF}}(u,v) = \sum_{l=1}^{l=\infty} \beta^l \left(\sum_{z \in \text{Path}_{uv}^l} \frac{|\phi(z)|}{\log d_z} \right), \quad (9)$$

where β is a parameter between 0 and 1. This parameter is used to control the contribution of path to the similarity; the longer the path is, the less contribution the path made to the similarity. $|\phi(z)|$ is the degree of node z in the subgraph containing the test nodes and all the path nodes between the test nodes. d_z is the degree of node z. The formula first calculated the guidance capability of each node on the path between node pairs and then put the sum of all path nodes' guidance capability as the contribution of the path to link prediction.

With the previous formula, we can give the pseudocode of KatzGF algorithm (see Algorithm 2).

The computational complexity of KatzGF algorithm is $O(K^n)$; K is the number of nodes in social network. It can be seen that when the network size is large, the time complexity degree of the KatzGF algorithm is very high. And KatzGF algorithm needs to know the global information of social networks. This is also difficult to achieve in real life, so KatzGF algorithm is not suitable for large-scale network.

5. Link Prediction Algorithms Based on Node Multiple Attributes

Traditional link prediction algorithm only intercepts a time snapshot of the social network, ignoring the time characteristics of the network. In fact, social network changes constantly over time; it is not static.

Considering the time characteristics of the network, social network graph can be divided into different graph sequences in accordance with a certain time step $G = \langle G_{\Delta t_1}, G_{\Delta t_2}, \ldots, G_{\Delta t_n} \rangle$. These snapshots graphs are disjoint. And $\{\Delta t_1, \Delta t_2, \ldots, \Delta t_n\}$ is a set of disjoint time. For all i, j, $1 \le i \le j \le n$, Δt_i is earlier than Δt_j. The selection of time step depends on the application. The time step can be set to a month or a year.

In order to illustrate the role of the node time statistics on link prediction better, we can make an analogy of the moving average line of finance. We use the moving average line (moving averages) to extract long-term development trends from the short-term noise data in finance. The moving average is the average of an index value in a certain period time. For example, we consider the average degree value of node v_i within 50 time steps; the average node degree is $\sum_{t=1}^{t=50} \text{degree}_t(v_i)/50$. In social networks, we can remove some

Input: social network graph $G = \langle V, E\rangle$, node x, node y
Output: the similarity of node x and node y
The description of the algorithm:
(1) Find the common neighbor set xy.common_neighbor of the node pair.
(2) Extract the sub-graph which contains the tested node pair and their common neighbors.
(3) While (the common neighbor ser is not null){
(4) Calculate the degree of node v, and get v.degree. Node v is one node of the common neighbor set.
(5) Calculate the degree of node v in the sub-graph extracted in the Step 2, get v.common_degree.
(6) Calculate the guidance capability of node v, Guidance (v) = v.common_degree/ log(v.degree)
(7) The similarity of node x and node y is Similarity.xy+ = Guidance (v)}

Algorithm 1: CNGF algorithm.

Input: social network graph $G = \langle V, E\rangle$, node x, node y
Output: the similarity of node x and node y
The description of the algorithm:
(1) Find all paths between node x and node y which length are less then 6. Put all the nodes in the paths into the set xy.path_nodes. Record the length of each path.
(2) Extract the sub-graph which contains the tested node pair and all the nodes in their paths.
(3) for (from the first path to the last path){
(4) while (the nodes set of the path is not null){
(5) Calculate the degree of node v, and get v.degree. Calculate the degree of node v in the extracted sub-graph in the Step 2, get v.path_degree.
Calculate the guidance capability of this node
Guidance-capability. v = v.path_degree/ log(v.degree).
(6) Calculate the weight of this path Guidance_capability.path+ = Guidance_capability.v}.
(7) Calculate the similarity of node x and node y Similarity.xy+ = β^l Guidance_capability.path}

Algorithm 2: KatzGF algorithm.

noise data using the node attribute average value and get a stable long-term trend.

Combining the node guidance capability and the average of node's degree, we propose a link prediction based on node topology attributes and network time attributes. In this section, we make some modifications of the node guidance capability's definition and add the time attributes into the definition of node guidance capability. We get a graph sequences $G = \langle G_{\Delta t_1}, G_{\Delta t_2}, \ldots, G_{\Delta t_n}\rangle$ from the social network graph $G = (V, E)$, where each snapshot graph is an undirected graph. In each snapshot graph, we calculate $|\phi(z)_{\Delta t_n}|$ and $d_{z-\Delta t_n}$, respectively. Then use the idea of nodes moving average and compute the average degree of node in the entire network and the extracted subgraph. The node's average degree in the extracted subgraph is $|\phi(z)|_{\text{avg}} = \sum_{i=1}^{i=n} |\phi(z)_{\Delta t_i}|/n$. The node's average degree in the entire graph is $d_{z-\text{avg}} = \sum_{i=1}^{i=n} |d_{z-\Delta t_i}|/n$. Finally, putting $|\phi(z)_{\text{avg}}|$ and $d_{z-\text{avg}}$ into the definition formula of node guidance capability, we can get a new formula that combines the time attributes:

$$\text{Guidance} - \text{force}(z)_{\text{time}} = \frac{|\phi(z)|_{\text{avg}}}{\log\left(d_{z-\text{avg}}\right)}. \quad (10)$$

With the previous new definition, we can modify the CNGF algorithm and KatzGF algorithm. We combine CNGF algorithm with social network topology attributes and time attributes and then get CNGF_T algorithm. The new similarity formula is

$$\text{Similarity}^{\text{CNGF\_T}}(u, v) = \sum_{z\in\Gamma(u)\cap\Gamma(v)} \frac{|\phi(z)|_{\text{avg}}}{\log\left(d_{z-\text{avg}}\right)}. \quad (11)$$

We also can get the new similarity formula in KatzGF_T:

$$\text{Similarity}^{\text{KatzGF\_T}}(u, v) = \sum_{l=1}^{l=\infty} \beta^l \left(\sum_{z\in\text{Path}_{uv}^l} \frac{|\phi(z)|_{\text{avg}}}{\log\left(d_{z-\text{avg}}\right)} \right). \quad (12)$$

6. The Algorithm Implementation and Verification

In this experiment, we used DBLP (Digital Bibliography and Library Project) experimental data sets to verify the performance of the link prediction algorithm. This data set is real. It covers 28 international conferences records related to data mining, machine learning, and so on.

We use AUC indicators to evaluate the experimental results. The AUC index refers to the area under ROC curve. ROC curve puts false positive rate as x-axis and puts true positive rate as cases y-axis. The point in the curve corresponds to the performance of the algorithm under each threshold. The greater the AUC value is the better performance the predictors have.

6.1. Data Preprocessing. The real conetwork is particularly sparse. In order to achieve the best performance of the algorithm, we need to preprocess the data set. The preprocessing of the data set can be divided into the following steps to complete.

Step 1. Intercept the data from 2000 to 2006.

Step 2. Delete the independent writings node.

Step 3. Remove the high degree of complete graphs.

Step 4. Construct the known network. Using the data from 2000 to 2004, we can construct the known conetwork. Put the topology structure attributes and the time attributes as the training set. And put the data from 2005 to 2006 as the unknown network.

Step 5. Screen less the known network. We did not select all the nodes of the data, so we should screen less the appropriate link. If a node is deleted, then all the links on the node should be removed.

6.2. The Experimental Results. By setting different thresholds, we calculate the true positive rate and the false positive rate, respectively. Then we draw the corresponding ROC curve. Firstly, we give the ROC curve of the local algorithm common neighbor and the ROC curve of the improved algorithms CNGF and CNGF_T, as shown in Figure 3. The ROC curve shows the overall impact of different thresholds for link prediction. It can be seen from the figure that the performance of three algorithms is very similar in the beginning; however, with the increase of false positive rate, the true positive rate of CNGF_T is larger than CNGF algorithm and common neighbor algorithm. This shows that the prediction performance of the CNGF_T algorithm is the best of the three algorithms.

The ROC curves of Katz algorithm, KatzGF, and KatzGF_T are shown in Figure 4. From the figure, we can see that KatzGF algorithm is better than Katz algorithm, but it is not very obvious in the overall performance. But the ROC curve of KatzGF_T algorithm is significantly closer to the upper left corner than those of the first two algorithms. KatzGF_T algorithm takes into account the time attribute information and the global topology information of social network, so it has the best prediction performance.

7. Conclusion

Firstly, we introduce the concept of social networks and describe the basic nature of social networks: small world, scaling, and clustering features. Then we introduce the link prediction definition and link prediction algorithms. Comparing these link prediction algorithms, we analyse the existing problems of the traditional link prediction. For the problems in traditional link prediction on social networks, we proposed the improved CNGF algorithm KatzGF algorithm. And for the lack of static social network, we gave the link prediction algorithm based on multiple attributes. The experimental results show that the prediction performance of the improved algorithm is superior to that of the traditional link prediction algorithm.

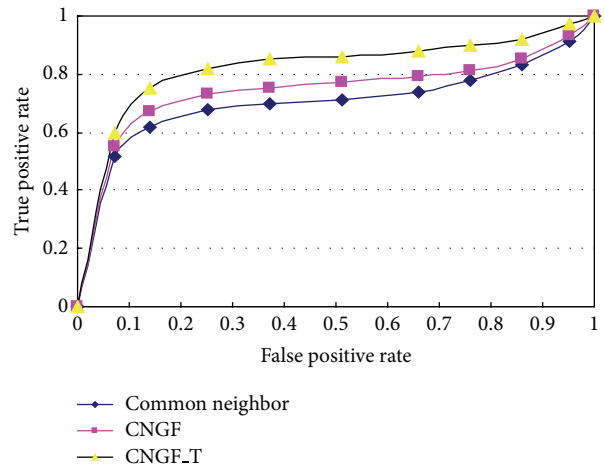

Figure 3: The ROC curves of common neighbor algorithm, CNGF algorithm, and CNGF_T algorithm.

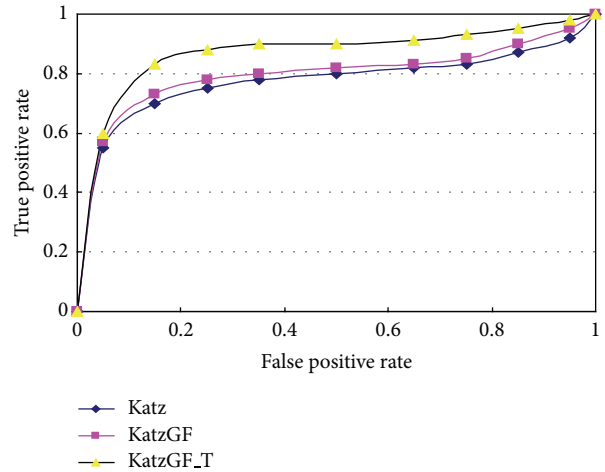

Figure 4: The ROC curves of Katz algorithm, KatzGF algorithm, and KatzGF_T algorithm.

References

[1] E. M. Airoldi, "Mixed membership block models for relational data with application to protein-protein interactions," in *Proceedings of International Biometric Society-ENAR Annual Meetings*, 2006.

[2] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, NK*, pp. 141–142, June 2005.

[3] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM Workshop of Link Analysis*, San Francisco, Calif, USA, 2006.

[4] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1994.

[5] J. Travers, "An experimental study of the small world problem," *Sociometry*, vol. 32, no. 4, pp. 425–443, 1969.

[6] R. Alberich, "Marvel Universe looks almost like a real social network," *Physieal Review Letters*, vol. 12, no. 4, pp. 12–18, 2006.

[7] G. Kossinets, "Effects of missing data in social networks," *Social Networks*, vol. 28, no. 3, pp. 247–268, 2006.

[8] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *American Association for the Advancement of Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[9] Y. D. Jin, T. Zhou, B. H. Wang, and B. Q. Yin, "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks," *Physical Review Letters*, no. 15, pp. 021–029, 2007.

[10] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[11] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[12] S. Brin, "The anatomy of a large-scale hypertextual Web search engine 1," *Computer Networks*, vol. 30, no. 1–7, pp. 107–117, 1998.