A link prediction algorithm based on trust and similar tag

Yu Jiankun
School of Information
Yunnan University of Finance and Economics
Kunming, China
yjk1102@163.com

Fan Sili
School of Information
Yunnan University of Finance and Economics
Kunming, China
540224846@qq.com

Abstract—the most similarity algorithm does not fully consider the network topology information, the paper design a calculation method of similarity, it takes the interrelation of nodes' tag and the text similarity of tag into account, most of trust calculation method is always treat all the nodes equally in the network, while ignoring the itself characteristics of each network node , and some trust computation method has certain subjectivity, This paper proposes a new method of calculating trust, finally, this paper presents a link prediction algorithm based on trust and similar tag (TAST)

Key words: similarity; trust; link prediction

I. INTRODUCTION

With the rise of social networking sites such as Sina micro-blog, the achievements of social network application attract people's attention, which makes the study on social networks emerge as the times require. Link prediction as a direction of social network analysis, many research achievements emerged. These studies can be divided into two categories, One is link prediction based on similarity, Lin was the first person to put forward the link prediction method that based on similarity in 1998, it use the attribute between nodes in the network to predict[1]. He believes that the similarity between the two nodes is higher, more possibility the two nodes have link, Newman et al proposed a common neighbor algorithm[2], It is a simplest similarity method based on local information,it put the number of contacts in common as the similarity index. Adamic proposed Adamic-Adar method[3], This algorithm was originally used to calculate the similarity of two user profiles, it has better performance. Katz proposed Katz method, This algorithm considers the network topology information, it considers all the path between the two nodes, and thus the amount of calculation is too large[4]. Another prediction method is based on trust. Kamvar et al proposed a model based on power iterations to compute the value of the trust ——EigenTrust[5], EigenTrust model must firstly select the starting point of trust, so Song proposed PowerTrust model to solve this shortcoming[6]. PowerTrust is a reputation system based on fuzzy reasoning. Golbeck et al established a trust model for the Semantic Web[7]. Actually, this model is an extension of the FOAF model[8]. It describes the specification and defines the appropriate trust body, and calculated the trust between users in trusted network by the principle of Six Degrees. To test this model, Golbeck developed a personalized movie review website——FilmTrust[9], FilmTrust background algorithm uses the trust value between users as the right to film scores, this can come to a personalized movie recommendations. Golbeck proposed a social reasoning trust algorithm TidalTrust , The main idea of this algorithm is that all users score their immediate friends, and calculating the trust of indirect friends. As the user score their Friend, so this algorithm has some subjectivity.

II. PROBLEM DEFINITION

There are many users $U=\{u_1, u_2,\ldots u_N\}$ and tags $I=\{i_1, i_2,\ldots i_M\}$ in a trusted network, we calculate the trust between two users who never had interaction by the original interactive in trust network, and we calculate the trust which generated by similarity of two users. In this process, we encountered two challenges: the first is the calculation method of interaction trust, the second is the user's choice of similarity label. These two issues seriously affect the precision and coverage of the recommendation system.

Let us analyze the existing problems through network structure drawing, Trust network that consisted by common neighbor nodes shown in Figure 1. Study on the possibility of nodes X, Y which generated link.

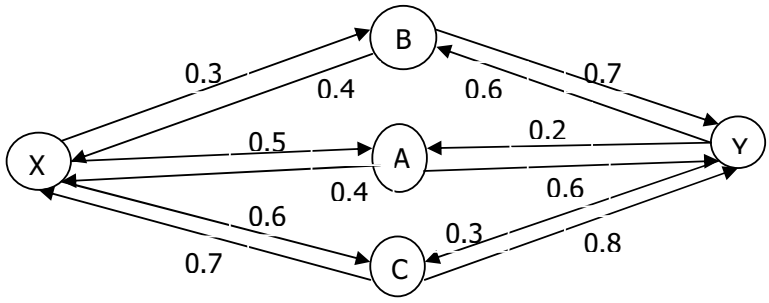

Fig. 1 diagram of the network

This research is funded by Business Intelligence Technology Innovation Team Foundation of Yunnan University

978-1-4799-6543-4/14 $31.00 © 2014 IEEE
DOI 10.1109/ICMeCG.2014.30

From the graph we can see, X, Y has an equal number of common neighbors (A, B, C), then we use the common neighbors algorithm to predict link, the possibility of X to Y and Y to X are equal. Meanwhile, if we use Adamic-Adar algorithm to predict link, since neighboring nodes A, B, C have equally degree, so the possibility of X to Y and Y to X are equal too. Thus, in Fig 1, the result of a common algorithm and Adamic-Adar neighbor algorithm is the same. If we use TidalTrust algorithm to predict link on Fig 1, the result of TidalTrust and common neighbors, Adamic-Adar is not equal. However, this algorithm is specifically certain subjectivity and arbitrary, the predicted results may not be a true reflection of reality.

III. TRUST CALCULATION METHOD

Many trust calculation methods, often equal treatment of all nodes in the network, these methods often have certain subjectivity and arbitrariness. Consider an graph G(V, E),with a set of nodes V and a set of edges E, where the size of nodes is n = |V |. Below, we use V_i to denote a node in a graph and v to denote a vector representation. The graph G can be represented as a adjacency matrix A, where A_{ij}= 1 if nodes (V_i, V_j) ∈ E otherwise A_{ij}= 0. The degree of a node Vi∈ G is denoted as $d_i = \sum_{j=1}^{n} A_{ij}$. Let D = diag(d_1, d_2, · · · , d_n) be a diagonal matrix of node degrees, so G can be defined using a transition matrix $P = D^{-1}A$. with entries $p_{ij} = \frac{A_{ij}}{d_i}$ .Consider a path τ_{ij}^{t} with length t from V_i to V_j. Suppose the path τ_{ij}^{t} passes through the nodes V_i , V_{i_1} ,… $V_{i_{t-1}}$, V_j , then the probability p(τ_{ij}^{t}) of the path τ_{ij}^{t} is defined as $p(\tau_{ij}^{t}) = p_{ii_1} p_{i_1 i_2} \ldots p_{i_{t-2} i_{t-1}} p_{i_{t-1} j}$. We can see this approach just put this direct trust degree is defined as the reciprocal of the number of the network nodes' out-of-degree, This is actually a binary trust network trust calculation method, and the actual situation of the network is not consistent.

For the above, this paper proposes a trust computing method that combining the actual situation of network. For TidalTrust arbitrary assignment of trust, we put the number of interactions between users as their trust weights, and we set a reduction factor based on the length of path. Therefore, we designed the method of the starting node S to trust the target node T as follows:

$$Tr(S,T) = \frac{1}{t}\sum_{i=1}^{n}[\prod_{j=1}^{t}\frac{NUM(V_j, V_{j+1})}{SUM_j}] \quad (1)$$

Where, t represents the number of nodes that from node T to node S. n represents the number of all paths that from node S to node T. SUM_j represents the number of V_j interact with all other nodes. NUM (V_j, V_{j+1}) represents the number of V_j interact with V_{j+1}. Meanwhile there is a case that there are three nodes A, B, C, nodes B, C is a direct neighbor of A, and the node B, C is a direct neighbor, then the number of the path from node A to node B is two: $A \to B$ and $A \to C \to B$. We can see the path $A \to C \to B$ is redundant. Because the nodes A, B already know, they did not need to interact through C. If using equation (1) to calculate the degree of trust in the trust network of Figure 1, we can get the trust of node X to node Y is 1/2*(0.3*0.7+0.5*0.6+0.6*0.8)=0.495, and the trust of node Y to node X is 1/2*(0.4*0.6+0.2*0.4+0.3*0.7)=0.265. We can see that the node X, Y trust not equal.

IV. SIMILARITY CALCULATION METHOD

We usually think that if two users attribute items are very similar, then the probability of they know each other is high. Some recommendation algorithm that usually only deal with the same attributes. Popular speaking, if a user he studied at the "XX university", recommended system recommend the user who belong to this university to this people. it did not deal with other universities which have related with this university.

In the current social network, users have a variety of tag attributes, users often sort this attribute, means that users of these tag attributes were rated. For the cases referred to in the preceding paragraph, the paper were processed the tags which is not exactly same. Here we used the pearson correlation coefficient [11] to deal with. pearson correlation coefficient ranges for [-1,1]. If it is negative, it means that two properties are completely different preferences. This situation is not useful to our study. Therefore, we consider only positive related items.

$$corr(i,j) = \frac{\sum_{u \in C_{i,j}} (r_{u,i} - \overline{r_u})(r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in C_{i,j}} (r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{u \in U} (r_{u,i} - \overline{r_u})^2}} \quad (2)$$

Where, $C_{i,j}$ indicates that the user collection who has a property item i and j, $\overline{r_u}$ indicates that the average score of user u rater all property. corr(i, j) indicates that the related degrees of the property item i and j.

The size of the set of common users is also important, For example, if corr(i, j) = corr(i, l), but $|C_{i,j}|>|C_{i,l}|$,then, since i and j have been rated by more common users, so the

correlation between them is stronger and sim(i, j) should be greater than sim(i, l). We consider $|C_{i,j}|$ in the similarity measure as follows:

$$sim(i,j)=\frac{\frac{1}{1+e^{-\frac{|C_{i,j}|}{2}}}\times corr(i,j)+similar(i,j)}{2} \quad (3)$$

We used the sigmoid function to avoid favoring the size of $|C_{i,j}|$ too much and to keep the similarity value in the range [0,1]. If the size of the set of common users is big enough, then the $\frac{1}{1+e^{-\frac{|C_{i,j}|}{2}}}$ part of equation (3) would converge to 1, but for small sets of common users, the factor $\frac{1}{1+e^{-\frac{|C_{i,j}|}{2}}}$ would be 0.6. $similar(i,j)$ is the cosine similarity function, which used to compute text similarity between the two tags. Therefore, the equation (3) consider the relationships of users, and consider the similarity between the users' tag, whereby the similarity between the user closer to the actual.

V. THE ALGORITHM PROCESS

Many of the current link prediction systems, it usually only deal with the same attribute items. Popular speaking, if a user he studied at the "XX university", recommended system recommend the user who belong to this university to this people. For this shortcoming, we propose A link prediction algorithm based on trust and similar tag, the algorithm process as follows:

Let the originating user u_0, and the target user u, the target tag i, then the algorithm process is calculated as follows:

```
1 FOR all users
2       RETURN Using Equation (1) to calculate the trust
        between user u0 and user u
3 END FOR
4
5 FOR  all target users
6       IF the tag of u and u0 is exactly same
7                 RETURN 1
8       ELSE
9                 RETURN Using
        equation(3) to calculate the
        similarity between the tag i(the tag
        of u0) and the tag j(the tag of u)
10 END FOR
11 ADD(the trust between u0 and u, the similarity of u0
    and u) // add the trust and the similarity
12 SORT(the trust) //sort the trust ,get Top-N users
```

In this process, the first step is to calculate the trust between originating user u_0 and target user u(1-3), then calculating the similarity between tag i(the tag of u_0) and tag j(the tag of u)(5-10), Finally, adding the trust and the similarity(11), then sort these value(12).

VI. EXPERIMENTAL RESULTS AND ANALYSIS

This paper designed a crawler used to obtain user information on Weibo as our experimental data (Figure 2,3 is an example of data). On the basis of these data, respectively experiment TAST algorithm, TidalTrust algorithm and common neighbor algorithm. We use recall, precision and F-value as an indicator to measure the performance of the algorithm. Recall that is coverage, assuming the trust network has n nodes, we predict the link between any two nodes. In this process, assuming the number of links that we predicted is N_P, and the correct number of links that we predicted is N_C, and the total number of links that actually generated is N_A, then the Coverage is N_C/N_A. The ratio of N_C and N_P is Precision. The F value is a composite indicator, it is $F\ Measure=\frac{2\times \text{Precision}\times Coverage}{\text{Precision}+Coverage}$.

user_id	screenName	is_vip	gende	interest	location	company	university	profile	tags	friend	follo	sta
176598018	Nic先森求人后	false	m	动漫/音乐/文	广东，深圳	(Null)	育才三中	爱奶茶爱咖啡	TF/OA/佐	173	851	86
271546854	阿敏小盆友	false	f	80后/吃货/小	上海，闵行区	普陀区中心	上海交通大	有种默契叫	偶稀饭腿	20	36	5
191637914	舒晓恩7289	false	f	后宫野史/	陕西，西安	(Null)	西安欧亚学	细节决定成败	学生/听歌	361	111	15
174465370	王永芳Amy	false	f	宅/模特/偶喜	上海，长宁区	(Null)	苏州工艺美	每个人都有自	水瓶座/艺	394	550	23
158866547	Kevin仔GZ	false	m	摄影/旅游/文	广东，广州	建衡(广州)	广东技术师	Better Me	旅行/处女	147	262	8
250306691	驴小毛Mo	true	m	射手座/驴小毛	北京，东城区	小脸通红工	(Null)	速东漫画，驴小毛/		146	4175	28
167839312	小丹梅花	false	f		黑龙江，哈尔	(Null)	(Null)	腿腿，我很喜	(Null)	60	95	2

Fig. 2 user data

id	user_id	web_id	comment_id
1	1025887503	3663021736433444	1912020577
2	1025887503	3663021736433444	1725214627
3	1025887503	3663021736433444	1591501991
4	1025887503	3663021736433444	2269639887
5	1025888104	3663022738978857	1322722721
6	1025888104	3663022738978857	1692590283
7	1025888104	3663022738978857	3609212955
8	1025888104	3663022738978857	1750472641
9	1025888104	3663022738978857	1453216515

Fig. 3 user interaction data

We selected 5000 user information, various interactive information that including attribute information to experiment. Below we show the number of links that between 5000 users.

TABLE I the number of links in 2013/9—2013/12

time	The number of users	The number of links	The number of new interaction
2013-9	5000	128483	0
2013-10	5000	133638	4410054
2013-11	5000	144042	5185512

2013-12	5000	125295	3382965

The number of new interaction in table 1 is the sum of users' message, reply and forwarded. Figure 4, 5, 6 shows the performance of three algorithms.

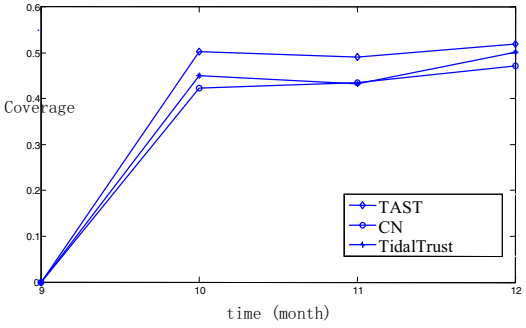

Fig. 4 Comparison of three algorithms of coverage

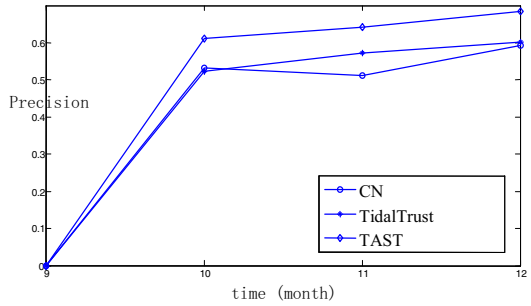

Fig. 5 Comparison of three algorithms of precision

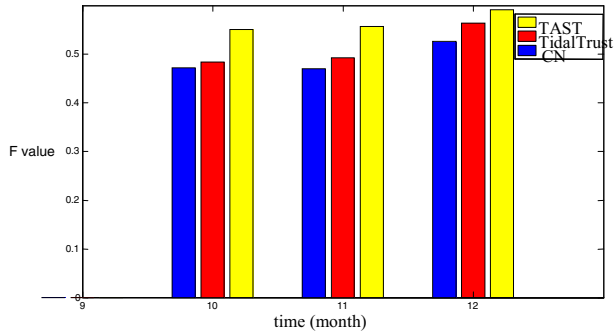

Fig. 6 Comparison of three algorithms of F-value

From the above three figure we can see, Common neighbor algorithm does not take advantage of the structure of networks and does not take full advantage of the node attribute information, moreover, without considering the interaction between neighbors and did not distinguish between the weight of each neighbor, it leads to poor performance. Meanwhile, TidalTrust algorithm is a link prediction algorithm based on trust, which uses a breadth-first algorithm to search node, the algorithm is superior to the common neighbor algorithm, but this algorithm has a certain degree of subjectivity, not very scientific, making it's prediction performance than common neighbor algorithm, but the performance of the algorithm inferior TAST. TAST algorithm takes advantage of structural characteristics of the network, it uses the number of interactions between the nodes as weights to calculate the degree of trust, meanwhile, it combine the relationship of user's tag with the similarity of user's tag, This makes it outperforms TidalTrust algorithms and common neighbor algorithm.

VII. CONCLUSION

Link prediction algorithm Based on the similarity and link prediction algorithm based on trust have some problems, and traditional link prediction algorithm based on trust have some shortcoming in treatment strategies, In view of this situation, this paper proposes a method for predicting links based on trust and similar tag. This approach is more realistic than TidalTrust algorithms and common neighbor algorithm, simulation results also verify the rationality and validity of this method.

Since the node number is too large in social network, which makes the computational complexity is considerable. Therefore, under the premise does not affect the performance of the algorithm, reduces the time complexity and space complexity will be my future research directions.

REFERENCES

[1] Lin D. An information theoretic definition of similarity[C]//ICML. 1998, 98: 296-304.

[2] Newman M E J. The structure and function of complex networks[J]. SIAM review, 2003, 45(2): 167-256.

[3] Adamic L, Adar E. How to search a social network[J]. Social Networks, 2005, 27(3): 187-203.

[4] Katz L. A new status index derived from sociometric analysis[J]. Psychometrika, 1953, 18(1): 39-43.

[5] Kamvar S D, Schlosser M T, Garcia-Molina H. The eigentrust algorithm for reputation management in p2p networks[C]//Proceedings of the 12th international conference on World Wide Web. ACM, 2003: 640-651.

[6] Song S, Hwang K, Zhou R, et al. Trusted P2P transactions with fuzzy reputation aggregation[J]. Internet Computing, IEEE, 2005, 9(6): 24-34.

[7] Golbeck J, Parsia B, Hendler J. Trust networks on the semantic web[M]. Springer Berlin Heidelberg, 2003.

[8] Brickley D, Miller L. The Friend of a Friend (FOAF) project[J]. 2000.

[9] Golbeck J, Hendler J. Filmtrust: Movie recommendations using trust in web-based social networks[C]//Proceedings of the IEEE Consumer communications and networking conference. University of Maryland, 2006, 96.

[10] Golbeck J A. Computing and applying trust in web-based social networks[J].University of Maryland, college Park,MD,2005.

[11] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.