A Friend Recommendation System Using Users' Information of Total Attributes

Zhou Zhang[1], Yuewen Liu[1], Wei Ding[2], Wei Wayne Huang[1]
[1] *Department of Management*
Xi'an Jiaotong University
Xi'an, China 710049
{zhouzhang@stu.xjtu.edu.cn, liuyuewen@mail.xjtu.edu.cn, whuang@mail.xjtu.edu.cn}
[2] *Department of Computer Science*
University of Massachusetts Boston
Boston, MA 02125
ding@cs.umb.edu

Abstract

Social network services, such as Facebook and Twitter in U.S.A., RenRen, QQ and Weibo in China, have grown substantially in recent years. Friend recommendation is an important emerging social network service component, which expands the networks by actively recommending new potential friends to users. We introduce a new friend recommendation system using a user's information of total attributes and based on the Law of total probability. The proposed method can be easily extended according to the number of user's attributes in different social networks. Our experimental results have demonstrated that superior performance the proposed method. In our empirical studies, we have observed that the performance of our algorithm is related with the number of user's friends. Our findings have important and practical applications in social network design and performance.
Key words: social network, common neighbors, friend recommendation, user's attributes

1. Introduction

Social network services, such as Facebook and Twitter in U.S.A., RenRen, QQ and Weibo in China, have grown substantially in recent years. Friends recommendation is crucial for the growth of social networks. At the early stage of social networks, the network is small with only a few users, it is easy to browse over other users' profiles to make a friend request. Nowadays, the number of social network users reaches an unbelievable level. In October 2012, the number of users in Facebook reaches one billion. The RenRen also have more than 200 million users by the end of 2012. Now it is obviously impossible for the user to browse over millions of other users' homepages to look for someone can be his/her friend. Social network users need an efficient friend recommendation system. For example, "People You May Know" of Facebook and other similar recommendation service are provided by Twitter, QQ, Weibo, and RenRen.

Existing friend recommendation algorithms in principle are based on two different approaches including the Path-based method and the Friends-of-Friend method. The Path-based method uses friend linkage information using concept of the well-known PageRank algorithm. Due to the high computational cost, this type of algorithms is seldom used in commercial social network services. The Friend-of-Friend (FoF) is an efficient and widely used recommendation algorithm in social networks due to its low time complexity. The algorithm identifies potential but unlinked friends and makes recommendations. Existing FoF algorithms only focus on the relations between users, but overlook user attributes.

In this study, we systematically evaluate the state-of-the-art algorithms to discuss their strengths and weaknesses. We then propose a new friend recommendation with user's information of total attributes (FRUITA). This paper is the first study that presents a friend recommendation system integrating social network users' attributes with the law of total probability. FRUITA can

be easily extended to accommodate new set of user attributes in different social networks. In our empirical study, we have extensively evaluated the FRUITA algorithm with other state-of-the-art FoF algorithms, including Common-Neighbors algorithm, Jaccard algorithm and Adamic/Adar algorithm using real-world data. We have collected 7 million users' public information and their friend relationships from one of China's dominant social network website. We have observed that the performance of our algorithm is related with the number of user's friends. In particular, when a user has a small number of friends, the proposed FRUITA algorithm performs much better than other algorithms; when a user has a large number of friends, the overall performance of FRUITA becomes less impressive but it is comparable with others and its precision rate is outstanding.

The rest of this paper is organized as follows. Section 2 gives a brief literature review of recommendation algorithms. Section 3 presents the methodology of the new algorithm. Section 4 discusses our real-world case study. Section 5 concludes the paper.

2. Related Literature

Recommending people is an important issue in social network. It has been shown that a recommendation service increases the connections between users, as well the user's loyalty to the social network. Different from recommending items, recommending people is relatively new in the research of social network, and there is less literature in this field. Friend-of-Friend and Path-based approaches are two basic methods.

(1) Friend-of-Friend (FoF) method

The FoF algorithm derives from the fact that if two users in the social network share many common friends, they may have a great chance to become friends in the future. This algorithm is also called as "Common-Neighbors". Newman designed an experiment and exploited the data of authors in two databases for a six-year period to provide evidence for the primary idea of FoF [4]. Their research also showed the proportional relation between the probability of the author having new co-authors and the number of the coauthors he or she already had. Jin et al. used the FoF algorithm as one of the three general principles to create a simple model that described the growth of social networks [5]. The friend recommendation system on Facebook, which gives a list of the "people you may know", is also based on the FoF algorithm.

As the continuous growth of social networks, the primary Common-Neighbors model proliferates into several improved algorithms, such as Jaccard coefficient and Adamic/Adar. In order to prove that some factors perform better in the link prediction problem, Adamic and Adar introduced a new algorithm to calculate the similarity of two actors by analyzing text, in-links, out-links and mailing lists on the homepages of the social networks [6]. The number of common friends between two actors can be used to evaluate the similarity.

Preferential attachment is one of well-known models to describe the expansion of social networks. Barabasi & Albert explained that a social network expanded when new actors joined in, and these new actors link preferentially to the old actors who have more links already[7]. Barabasi et al. (2001) studied the data with an 8-year period in a database of co-authorship information, and tried to find the evidence of preferential attachment in the evolution of social network [8].

(2) Path-based method

Differing from the neighbor-based FoF approach, calculating the shortest path is the basic idea of the Path-based methods. Katz predicts the probability by the sum of all paths between two nodes. And the shorter paths have more contribution than the longer paths in the link prediction[9].

Brin & Page introduced the PageRank algorithm as a key component of Google search engine. It weighs every element within a set by the link-in and link-out numbers, and then gives a rank of all the elements [10]. There are several improved algorithms based on PageRank [11, 12].

Jeh and Widom proposed SimRank to measure similarity of elements using the information of their relations. SimRank combined the features of FoF and the Random Walk algorithms, and Random Walk is also used in PageRank [13].

Yin proposed and evaluated a framework of LINKREC, which used the information of the network structure and the actors' attributes, based on the Random Walk with Restart algorithm[14].

3. Methodology

For a friend recommendation system, an example of a candidate friend may be $\langle x_1, x_2, \cdots, x_i, \cdots, x_m \rangle$. $x_i \left(i \in \{1, \cdots, m\} \right)$ stands for the attributes of the candidate, such as gender, age, location, interest and number of common-neighbours, these attributes may be independent or not. For example, young men may show strong interest in sports, so the gender and age will actually have influence on the attribute of interest. Even if some of the attributes are not independent, we still use Equation (4) to calculate the total probability of friend recommendation under strong independence assumption. Because we don't use the calculated probability value to directly predicate the chance that the candidate will really become a friend of the user in the future, we just use the probability values to select potential strong candidates. Our friend recommendation system will give the user a list of candidate friends ranked by the probability values. The advantage of decoupling of the class attributes using the strong independence assumption is that we can independently calculate each user attribute distribution quickly. Similar as the theory behind naïve independence assumption used in the successful naïve Bayesian classifier [23], dependence among users' attributes may likely be canceled out, and the performance of our friend recommendation system can still be strong. Our empirical results have approved our argument.

For each attribute, we can calculate the prior probability by the data of the existing friends of the user. The relation between a candidate and the user can only be two types: friends or not. Let *y* indicates a binary variable which reflects the relation between the candidate and the user. If the candidate is a friend of the user, we define *y* = 1; else *y* = 0. Consider $x_i \left(i \in \{1, \cdots, m\} \right)$ as the attributes of the user, then the probability that the user will collaborate with the candidate is:

$$P\left(y = 1 | \bigcap_1^m x_i \right) = 1 - \prod_1^m \left(1 - P\left(y = 1 | x_i \right) \right) \tag{1}$$

In Equation (1), *m* denotes the number of user's attributes existing in the social network. $P(y = 1 | x_i)$ denotes the prior probability for each attribute that the probability that this candidate will be friend of the user in the future. It can be calculated by the statistical result including the information of all the friends of the user's existing friends (friends-of-friend) and the number that how many of them are already friends of the user. $\prod_1^m (1 - P(y = 1 | x_i))$ denotes the probability that the candidate will not be the user's friend based on all the *m* attributes.

Algorithm 1: FRUITA (Friend Recommendation with Users' Information of Total Attributes)

1. Input: The database of the friendship relations between users in the social network; the database of the users' *m* attributes.
2. Construct the social network graph for the user by the database of the relation. All the friends of the user's existing friend are V_t; the set of the persons in V_t who have already been friends of user is V_f; the set of the other *n* persons in V_t will be the candidates for the friend recommendation system and we mark it as V_c.
3. Estimate the probability $P(x_1)$ that V_t will be friend of the user for attribute i by the statistical result of V_t and V_f. For all *m* attributes, we will get $\{P(x_1), P(x_2), \ldots, P(x_m)\}$.
4. Calculate the probability *P* for each of the *n* candidates in V_c using Equation (5) and $\{P(x_1), P(x_2), \ldots, P(x_m)\}$.
5. Sort the *n* candidate by the value of probability *P*.
6. Return: Top *k* of the sorted *n* candidates as the list of friend recommendation result.

The pseudo-code of recommendation algorithm FRUITA is shown in Algorithm 1. In step 3, if calculating each P of the attribute costs time m and there are n attributes, the time complexity of step 3 is $O(mn)$; in step 4, if calculating each P of the candidates costs time m and there are n candidates, the time complexity of step 4 is $O(mn)$; in step 5, we use the function "Rank()" in SQL to sort the results and the time complexity of step 5 is $O(n \log n)$.

4. Empirical Study

In order to carry out the experiments, we use a web crawler to get the user data from RenRen (http://www.renren.com) and store it into a database. RenRen is one of the most popular social network websites in China and have more than 200 million users in total. First, we download the information of 240 users with different attributes and we defined them as D1 nodes. Second, we extend to the information of 51,340 D2 nodes which are the friends of these 240 users. Third, we keep on collecting the data of the D2 users' friends and we call them D3 nodes and there are 7,158,934 D3 in total. These nodes and the edges between them form a social network structure for our case study.

With the data we get from RenRen, we have evaluated FRUITA with other state-of-the art FoF algorithms. Specifically we split each user's friends to 10 partitions, and try to see how well one specific algorithm can predict 1 partition using the other 9 partitions. As depicted in Figure 3. This method of handling the data collected in a time point is widely used in the field of friend recommendation in a social network. This method also has one significant limitation. The friend recommendation results that are not in the set of the 1 partition do not mean they are wrong, because some of them may be the potential friends of the user and will be added by the user as friends in the future. So we expect that the actual precision value of the algorithms should be higher than the value in the evaluation report.

The link prediction results are showed in Tables 1-3.

Table 1 shows an overall result of the friend recommendation for the 240 D_1 users in RenRen. We can see that the FRUITA performs best in MAP (16.97%), and some P@N (76.92% precision at 1, 50.17% precision at 2, and 10.83% precision at 100). Common-Neighbors and Adamic/Adar perform well too. Their MRRs are 40.51% / 41.59% and MAPs are 16.24% /15.97%, both comparable to FRUITA. The result of Jaccard's coefficient is acceptable, but worse than other three.

Table1 overall result of algorithms comparison

	P@1	P@2	P@5	P@10	P@50	P@100	MRR	MAP
FRUITA	**0.7692**	**0.5017**	0.3897	0.2823	0.1719	**0.1083**	0.4121	**0.1697**
CN	0.6581	0.4957	**0.3932**	**0.2908**	**0.1737**	**0.1083**	0.4051	0.1624
JAC	0.5000	0.4171	0.3436	0.2675	0.1649	0.1069	0.3736	0.1340
ADA	0.6154	0.4744	0.3782	0.2812	0.1679	0.1076	**0.4159**	0.1597

Then we divide the D_1 users by the number of their friends into two groups, and repeat the experiments. Table 9 shows the result of the D_1 users whose friends are less than 100, and Table 10 shows the result of the D_1 users whose friends are more than 100.

In Table 2, all the results are worse than Table 1 as expected. The FRUITA has the best MAP (20.69%), P@50 (2.95%). The Common-Neighbors has the best P@1 (40.68%), P@2 (16.27%), P@5 (10.51%), and P@10 (6.36%). The result of Adamic/Adar is not as good as Common-Neighbors and FRUITA, but still comparable. The result of Jaccard's coefficient is much worse than other two algorithms and unacceptable.

In Table 3, all the results are better than Table 1. The Common-Neighbors beat other three algorithms in most of the indices (MRR 44.17%, P@5 49.03%, P@10 36.74%, P@50 22.29%, and P@100 13.95%). The result of FRUITA is impressively outstanding on P@1 91.43% and P@2 61.71%. Because the top recommended person is always the first one browsed by the user, P@1 is the most important one in P@k. The results of Adamic/Adar are comparable to FRUITA and Common-Neighbors. Jaccard's coefficient is still worse than the other three, but the gap is evidently narrowed than the value in Table 5.

Table 2 result of algorithms comparison (Friends <100)

	P@1	P@2	P@5	P@10	P@50	P@100	MRR	MAP
FRUITA	0.3390	0.1593	0.1000	0.0627	**0.0295**	0.0164	0.3287	**0.2069**
CN	**0.4068**	**0.1627**	**0.1051**	**0.0636**	0.0281	0.0158	0.2963	0.1739
Jaccard	0.1186	0.0610	0.0492	0.0305	0.0183	0.0112	0.1901	0.0997
Ada	0.2373	0.1288	0.0847	0.0576	0.0281	**0.0169**	**0.3430**	0.1530

Table 3 result of algorithms comparison (Friends >100)

	P@1	P@2	P@5	P@10	P@50	P@100	MRR	MAP
FRUITA	**0.9143**	**0.6171**	0.4874	0.3563	0.2199	0.1393	0.4402	0.1572
CN	0.7429	0.6080	**0.4903**	**0.3674**	**0.2229**	**0.1395**	**0.4417**	0.1586
Jaccard	0.6286	0.5371	0.4429	0.3474	0.2143	0.1391	0.4350	0.1456
Ada	0.7429	0.5909	0.4771	0.3566	0.2151	0.1382	0.4404	**0.1619**

Our extensive empirical studies have shown that (1) in total, FRUITA performs much better than other basis algorithms. The performances of Common-Neighbors and Adamic/Adar algorithms are better than Jaccard's coefficient; (2) When the user has relatively less friends (e.g., <100), FRUITA performs better than Adamic/Adar and Common-Neighbors, and much better than Jaccard's coefficient; (2) When the user has relatively more friends (e.g., >100), the performance of FRUITA, Common-Neighbors and Adamic/Adar performs are comparable, and Jaccard's coefficient is still the worst. The precision of FRUITA is impressively outstanding at top recommended results.

5. Conclusions

The FRUITA not only inherits the advantage of FoF but also has a flexible format which can be easily extend according to the number of user attributes. We evaluate the new algorithm with other FoF algorithms using real-world data. Our result shows that the FRUITA performs best of all in total. And our study also finds that performance of all these friend recommendation methods may depend on the number of users' existing friends. When the number of existing friends is falling down to less than 100, the result of Jaccard's coefficient may be unacceptable and Adamic/Adar also performs worse but still acceptable. By contrast, Common-Neighbors and FRUITA keep perform well. Furthermore, FRUITA still keep its strong performance when the number of existing friends increases, while other algorithms may not be able to do so.

We also observed that the way of utilizing information is very important for an algorithm. Adding extra information to an algorithm does not necessarily enhance the performance of the algorithm, unless the information is integrated properly. The Common-Neighbors algorithm utilizes only the number of common-neighbors; the Jaccard's coefficient utilizes more information, including the number of common-neighbors, the number of the user's and the candidate's friends, but ironically performs worse than the Common-Neighbors algorithm, because the three numbers are integrated arbitrarily rather than properly. The Adamic/Adar algorithm also utilizes more information, i.e., the number of friends of the common neighbors. However, when the number of common-friends is relatively low, introducing extra information to the algorithm may introduce too much noise, thus the Adamic/Adar algorithm performs not better than the Common-Neighbors algorithm. When the number of common-neighbors is relatively high, the noise brought by the number of friends of common-neighbors is weakened, thus the Adamic/Adar algorithm performs better than the Common-Neighbor algorithm. Compared to Adamic/Adar, FRUITA efficiently utilizes users' information. It can handle all the user attributes flexibly in a social network. And the recommendation results will be enhanced with the increase of the number of user's attributes.

References

[1] M. Pazzani and D. Billsus, "Content-based recommendation systems," *The adaptive web*, pp. 325-341, 2007.

[2] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, pp. 393-408, 1999.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 734-749, 2005.

[4] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, p. 025102, 2001.

[5] E. M. Jin, M. Girvan, and M. E. Newman, "Structure of growing social networks," *Physical review E*, vol. 64, p. 046132, 2001.

[6] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, pp. 211-230, 2003.
[7] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, pp. 509-512, 1999.
[8] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications*, vol. 311, pp. 590-614, 2002.
[9] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, pp. 39-43, 1953.
[10] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer networks and ISDN systems*, vol. 30, pp. 107-117, 1998.
[11] T. H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 15, pp. 784-796, 2003.
[12] T. Haveliwala, S. Kamvar, and G. Jeh, "An analytical comparison of approaches to personalizing pagerank," 2003.
[13] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 538-543.
[14] Z. Yin, M. Gupta, T. Weninger, and J. Han, "A unified framework for link recommendation using random walks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, 2010, pp. 152-159.
[15] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.
[16] 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*, 2005, pp. 141-142.
[17] D. Liben‐Nowell and J. Kleinberg, "The link‐prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, pp. 1019-1031, 2007.
[18] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *Proceedings of the 27th international conference on Human factors in computing systems*, 2009, pp. 201-210.
[19] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 1998, pp. 43-52.
[20] H. Zhang and J. Su, "Naive Bayesian classifiers for ranking," in *Machine Learning: ECML 2004*, ed: Springer, 2004, pp. 501-512.
[21] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, " Combining content-based and collaborative filters in an online newspaper," in *ACM SIGIR'99. Workshop on Recommender Systems: Algorithms and Evaluation***,** August 1999.
[22] I. Soboroff, and C. Nicholas. "Combining content and collaboration in text filtering," in *43 IJCAI'99 Workshop: Machine Learning for Information Filtering*, August 1999.
[23] H. Zhang, " The optimality of naive Bayes," in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, Miami Beach, AAAI Press, 2004.