

Recommender Agent Based on Social Network

Heng-Li Yang and Hsiao-Fang Yang

Department of MIS, National Cheng-Chi University,
64, Sec. 2, Chihnan Rd., Mucha Dist, Taipei 116, Taiwan
{yanh, 94356507}@nccu.edu.tw

Abstract. Conventional collaborative recommendation approaches neglect weak relationships even when they provide important information. This study applies the concepts of chance discovery and small worlds to recommendation systems. The trust (direct or indirect) relationships and product relationships among customers are to find candidates for collaboration. The purchasing quantities and feedback of customers are considered. The whole similarities are calculated based on the model, brand and type of purchased product.

Keywords: recommender system, social network, chance discovery, trust.

1 Introduction and Related Work

Internet technology is changing day by day, and has become an important media channel of enterprises in the twenty-first century. By transcending the limits of time and space, electronic commerce (EC) has changed business marketing and people life-style. However, high growth rates are difficult to obtain in highly competitive and turbulent environments. Maintaining the loyalty of current customers and further attracting new customers are challenges for business.

To attract customers to purchase products, recommendation systems are often embedded into EC web sites [14]. Common approaches include content-based approach [2,6] and collaborative filtering [1,4,13,15,16]. The content-based approach analyzes the content of item description documents, compares customer profiles, then recommends highly-related items to the customer. InforFiner [5] and NewsWeeder [6] are some examples of content-based systems. This approach has some limitations, e.g., the difficulties of analyzing multimedia documents, generating serendipitous finding, and filtering items based on criteria other than those disclosed [16].

Collaborative filtering is an alternative approach, which recommends items that other similar users have liked. It calculates the similarity of users rather than the similarity of items. Ringo [16] is an collaborative filtering application for recommending music.

To judge similarity, some clusters of customers with the same characteristics (e.g., shared experiences, location, education or interests have to be found. However, this study argues that the clustering approach has some blind spots in. First, a customer might not trust the recommendations of customers in a cluster assigned by the system. Second, two customers in different clusters could still reference each other recommending *A* based on *B*'s preferences. This is the kernel spirit of chance

discovery: chances occur while mining low relationships but important information [9]. This study applies the idea of chance discovery to the recommendation approach.

However, the problem is locating these weak but important relationships. As shown in Fig. 1, relationships among customers of a commercial web site can be categorized into two groups. The first class of relationship, namely product, occurs because customers purchased the same product(s), e.g., the same model, brand or type. The second class of relationship is trust. Some previous studies have recommended trust-based recommendation systems [3,7,8,10,11,12]. For example, Pitsilis and Marshall [11,12] recommended establishing a direct trust link between two people with common experiences (i.e., purchased the same items), and deriving indirect trusts through a chain. Battiston et al. [3] suggested that when purchasing an item, consumers could query their neighborhood for recommendations. O'Donovan and Smyth [8] proposed a system that automatically infers trust values from ratings between users and movie producer. Massa et al. [7] suggested that consumer *A* could assert a trust statement toward *B* if *A* consistently found the product reviews and ratings of *B* valuable. Integrating the concepts of the above works, this study proposes checking the list of trustful customers, which are explicitly established by same customers. The listing of trustful customers might already exist before becoming the customers of the web. Alternatively, customers could provide names by searching product reviews or the forum discussions at this web site. People may consult the suggestions from others that they directly or indirectly trust when buying something.

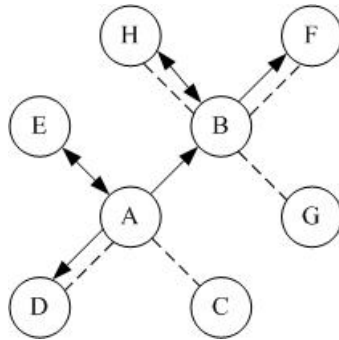


Fig. 1. Two kinds of networks among members: dotted line is product connection (purchase the same product), solid line is trust connection, $A \rightarrow B$ implies *A* trusts *B*

The number of relationships that should be consulted could be determined by the small world theory of developed by Watts [17]. The small world theory claims that the distance between any two people is no more than six degrees. People can be clustered into groups based on characteristics. A person may belong to several clusters simultaneously at the same time. Because of such overlap, a target person can be found by consulting only six persons. In the proposed model, the indirect trust listing could be constructed from the direct trust listing: *A* trusts *C* because *A* trusts *B*, and *B* trusts *C*. However, for efficiency, this study only considers three layers of trust.

The rest of this study is organized as follows. Section 2 explains the main idea of the proposed approach. Section 3 describes a scenario to explain how such an approach would help customers. Section 4 draws conclusions.

2 The Proposed Approach

The web site is assumed to maintain basic profiles of its members. The past year of detailed transaction data should be kept. A proper product type hierarchy should be maintained. These data enable the system to determine the product model, brand and type bought by each member. The granularity of type classification, fine or coarse, depends on management decisions. For example, the chain of ancestors of model “Nokia model N90” could be “Nokia model N-series”, “Nokia 3G cellular phone”, “3G cellular phone”, “cellular phone”, “telephone”, “communication product”, “electronic product”. The website should also maintain the evaluation scores of bought products for each customer. If a customer bought a product for one month, then the website would e-mail him (her) to ask for his (her) evaluation score (“1” to “7”). The website might give some bonus to encourage the replies. If the customer did not reply within some specified period, then a default score “4” would be assigned.

Additionally, the trustfulness listing of each member is obtained. It could be established at the time that a person became the member of the website, and could be updated anytime later by the member. The website has a forum for customer discussions and product reviews. The persons on the trustful list are also members of the website. They may be friends before in real world or have become acquainted in the virtual world. Alternatively, customers could pick up the member names after reading their product reviews. A member could only provide five names (IDs) for all products, or could give five names for each product type. However, even if the website already had the name listing, the system would ask when a member requesting a recommendation to confirm his trust rating.

The proposed approach has two phases. Based on the two relationships mentioned above, the first phase searches all candidate persons with similar purchasing experience. The second phase calculates the similarities of each candidate, and recommends the candidate with the most similar experience.

2.1 Candidate Search Phase

As shown in Fig. 2, the first phase would attempt to find candidates from trust and product relationships. Users are then requested to clarify the recommendation task: specifying product model series (e.g., “Nokia N-series cellular phone”, product brand (e.g., “Nokia 3G cellular phone”) or product type (i.e., “3G cellular phone” or “cellular phone”). As shown in Fig. 3, at most three layers of trust are consulted. In the first derivative ($i=1$), the direct trustful list (five persons at most), is applied. The transitive trusts are derived from the direct trusts: the second derivative ($i=2$) consists of 25 persons at most, while the third derivative ($i=3$) consists of 125 persons at most. The expansions stop once any candidates are found. As shown in Fig. 4, when consulting with the trusted persons, the system checks whether these trusted persons have purchased any related product model (e.g., any “Nokia”, “3G cellular phone” or “cellular phone” model) based on the product type hierarchy.

Procedure Search_Candidate

Step1: Initialize $i=1$ // i denotes the expanding times of trustful relationship

Step2: Initialize TL_0 and Candidate to be {} // TL_i denotes a trust list; TL_0 , the initial set, is empty; Candidate store all candidates who purchased the specified product

Step3: Set r as some specified product model series, brand, or type specified by the user or automatically recommended by the system.

Step4: Ask the user to confirm or provide the trustful list TL_1

Step5: Do while ($i \leq 3$) and Candidate={} // at most search 3 derivatives of trust list

Step6: Merge TL_{i-1} into TL_i ; sort the result, and delete any duplicates

Step7: for $k = 1$ to length(TL_i) // the number of trusted name are length(TL_i)

Step8(check-loop): for all sub-layers j of r in the type hierarchy

Step9: if purchased (k,j) = true then insert k into Candidate and exit the check-loop // k -th person purchased

Step10: end //end of steps 8-10 to check person k

Step11: end//end of steps 7-11 to check persons in TL_i

Step12: $i = i+1$ // advance to next derivative

Step13: end //end of steps 5-13, the while loop

Step14: End //end of the procedure

Fig. 2. The candidate search algorithm

If no candidate is found, then the system automatically relaxes the specification constraint to climb up one level of the type hierarchy, and calls the procedure *Search_Candidate* again. For example, if the website cannot find any trusted person who purchased “Nokia 3G cellular phone”, then it relaxes the constraint to “3G cellular phone”. If any candidate is found, after relaxing the constraints, then the system provides the information to the user, and discusses with the user whether to confirm such a constraint relaxation.

The procedure *Search_Candidate* has three loops. The outer loop (steps 5-13) is iterated at most three times. The middle loop (steps 7-11) is iterated 125 times in the worst case. The worst case of iteration in the inner loop (steps 8-10) depends on the

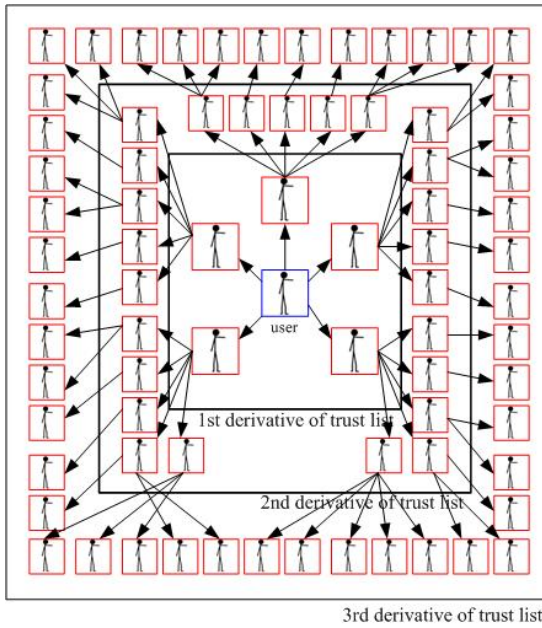


Fig. 3. The expansion of trustful list consultation

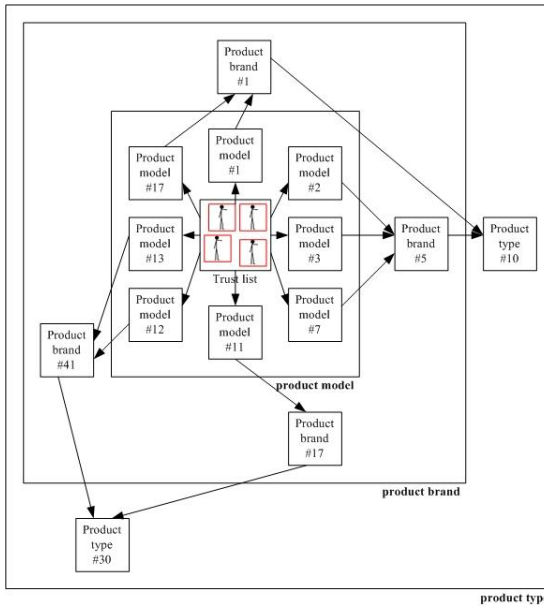


Fig. 4. The product hierarchy consultation

number of sub-items belonging to an item in the type hierarch. In the real case, this number is small. The merge-sort in Step 6 also takes a short time, because the maximal number of the list is only 125. Therefore, this algorithm is efficient with a good data structure.

The density of the social network is related to the performance of recommendation system [3]. If the network is dense enough, i.e., has “small-world” properties [17], then the recommender agent could reach all nodes in a few hops, enabling the customer to obtain a recommendation. This study ran a simulation by assuming 10 product types, where each type had 100 brands, each brand had 100 models, and there were 10000 members. Each member was given a purchasing record by randomly assigning the product models. Finally, the recommender agent was found to be able to give recommendations if each member had purchased two product models and provided one trusted member, or even better if they had purchased one product model and provided two trusted members.

2.2 Similarity Computation Phase

In the second phase, the website calculates the similarity between the query person and the candidates. The website would suggest the best product item evaluated by the person who is most similar to the query person. The similarity would be based on the whole purchase experiences — not only items, but also the quantities of the items, and the evaluation feedback from users. Furthermore, with regard to the product items, three levels (model, brand, primitive type) are considered together. The similarity of two users is certainly high if they purchased the same product models, but is also high even if they bought the same brands, but different models.

Assume that in the website the numbers of product model, brand and primitive types are M , B and T , respectively. The primitive types are those with one level higher than product brands in the type hierarchy. Additionally, suppose that the number of members is N . For efficiency, the website may periodically construct three feedback matrices, FM , FB , and FT . The dimensions of these matrices, FM , FB , and FT are $N \times M$, $N \times B$, $N \times T$, respectively. These matrices store the weighted average evaluation feedback scores of all members on all product models, brands and primitive types in this recent year, respectively. For example, if a user u purchased product model d three times, in quantities of 15, 20, 32, and gave the evaluation scores 6, 4, 5, respectively, then his weighed evaluation score on this product model is 4.9, i.e., $(6*15+4*20+5*32)/67$.

Formula (1), (2), (3) are used to compute the similarities between the query member x and any candidate u on the dimensions of product model, brand, and primitive type, respectively. Formula (4) determines the total similarity between these two users. The weighs w_1 , w_2 , and w_3 are applied to product model, brand, and type, respectively. If two persons bought the same product model, then they would buy the same brand and type consequently. Therefore, this study recommends assigning values such that $w_1 > w_2 > w_3$. The max function would return the maximal values of all values of SM, SB, or ST.

$$SM(x, u) = \sum_{m=1}^M |FM[x, m] - FM[u, m]| \quad (1)$$

$$SB(x, u) = \sum_{b=1}^B |FB[x, b] - FB[u, b]| \quad (2)$$

$$ST(x, u) = \sum_{t=1}^T |FT[x, t] - FT[u, t]| \quad (3)$$

$$S(x, u) = w1 \times \frac{SM(x, u)}{\max(SM)} + w2 \times \frac{SB(x, u)}{\max(SB)} + w3 \times \frac{ST(x, u)}{\max(ST)} \quad (4)$$

3 A Scenario

A person, James, is currently unsure about what notebook he should purchase for his work. He has not used any notebook before. Additionally, he is not familiar with the technology market. He is worried about the notebook price, performance and future possible repair support. If he is a member of the website, then he can use it to help him choose a suitable notebook. The user could first browse the product items and read the descriptions on the website. Suppose that he has good previous impressions of IBM products and has become interested in the Z-series after reading the product descriptions. The IBM notebooks are available in several series, such as X, T, R and Z. The Z-series has a number of models, such as Z60t and Z60m.

The user makes a recommendation request. The website first asks him to provide a list of names of trusted users, at most five members. If he cannot provide such a listing, then the website can only recommend the best Z-series model as evaluated by all of the members who had purchased before. Suppose that he provides the names of three members. In the candidate searching phase, the website checks whether these members have purchased any Z-series model. If no directly trusted member has such purchase records, then the system searches the indirect trustful member listings in at most two iterations. If neither direct nor indirect members have such experiences, then the website relaxes the constraints to set the search target to the higher level of product items (e.g., any series of IBM notebook). If the system can find any suitable candidate after relaxing constraints, then it gives the user a message such as “No members whom you trust directly or indirectly have purchased any IBM Z-series model notebook, but some have purchased IBM R-series models. Would you agree to relax your original constraints?”

If more than one candidate exists, then the system calculates the similarity, and recommends the purchase experiences of the candidate (say Tom), who was most similar to James from past experience. For example, the system would give the user a message such as “Based on the previous records, the system has found that among those your directly (or indirectly) trusted persons, as the whole, Tom, had purchase behaviors most similar to you. Among those IBM Z-series models, which Tom

purchased, he gave *Z60m* the highest feedback evaluation. Therefore, we suggest you consider *Z60m*". If James were a new member without any purchase records, then the system could not calculate the similarity. In that case, the system would suggest him the best Z-series model, which was evaluated by all of the members whom James trusted (of course, the directly trusted would be given first priority).

Figure 5 shows the GUI of a proposed recommendation system. On the left side, a user can browse product descriptions, input/modify trust lists and request recommendation via type hierarchy.

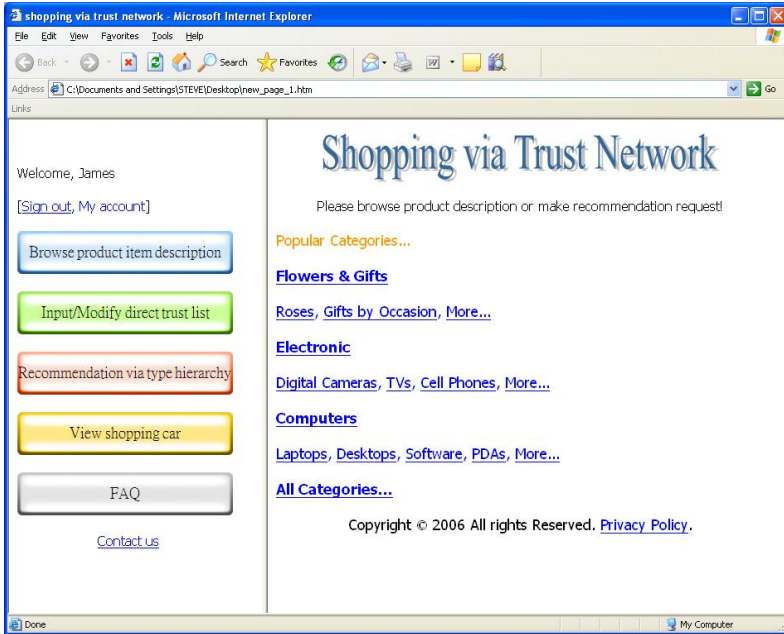


Fig. 5. The proposed recommendation system

4 Conclusions

Integrating the ideas of the literature, this study applies the ideas of chance discovery and small world to recommendation systems. The proposed approach has the following characteristics of our approach. (1)The explicit trust relationships are respected. The indirect trusts are derived beginning from the direct trust listing. The theory of chance discovery means that indirect trust relationships might imply low relationships, but provide important [9]. Watts [17] reported that in a world consisting of hundreds of thousands of individuals, every actor could be connected to every other actor within an average of less than six steps. This study uses three layers of trustful lists. Customers should feel comfortable because the consulted persons are within their trust networks, had all purchased the requested product, had the most similar purchasing experiences. Therefore, the recommender would pay attentions to

psychological status as well as the actual behaviors of the consumers. Suppose that a consumer John chooses a particular product brand because it has been purchased by Mary whom he trusts. However, upon consumption, he discovers that this brand does not match his taste and gives low evaluation feedback. In this case, the similarity score between John and Mary is lowered. John would be less likely to follow the recommendations of Mary the next time he goes shopping, although Mary remains in his list of trusted members. (2) In case the customer does not know any other members, he could pick up the professionals as the trusted persons after querying some of the product reviews and judging who are “best” buyers. Notably, trustful persons must first be selected to enable the customers to refer to their experiences. (3) The concept of product type hierarchy is used to allow the recommendation to climb up or drill down to search the “same” product purchases. The system relaxes the user’s specification and provides more intelligent suggestions. (4) The similarities between customers are based on overall pictures of their purchase behaviors: purchasing quantities, feedback of customers, the whole chain of product model, brand, and type. Therefore, the similarity would be integrated and holistic. Future work will be to implement the proposed approach, and perform experiments to evaluate its performance.

References

1. Balabanovic, M., Shoham, Y.: *Fab: Content-Based, Collaborative Recommendation*. Communications of the ACM. 40(3), 66–72 (1997)
2. Basu, C., Hirsh, H., Cohen, W.: *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*. In: Proceedings of AAAI Symposium on Machine Learning in Information Access (1998)
3. Battiston, B., Walter, F.E., Schweitzer, F.: *Impact of Trust on the Performance of a Recommendation System in a Social Network*. In: W Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS’06). Hakodate, Japan (2006)
4. Goldberg, D., Nichols, D., Oki, B., Terry, D.: *Using Collaborative Filtering to Weave an Information Tapestry*. Communications of the ACM. 35(12), 61–70 (1992)
5. Krulwich, B., Burkey, C.: *Learning User Information Interests through Extraction of Semantically Significant Phrases*. In: Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access (1996)
6. Lang, K.: *Newsweeder: Learning to Filter Nnetnews*. In: Proceedings of the 12th International Conference on Machine Learning, pp. 331–339. Morgan Kaufmann, San Francisco (1995)
7. Massa, P., Bhattacharjee, B.: *Using Trust in Recommender Systems: An Experimental Analysis*. In: Jensen, C., Poslad, S., Dimitrakos, T. (eds.) *iTrust 2004*. LNCS, vol. 2995, pp. 221–235. Springer, Heidelberg (2004)
8. Montaner, M., Lopez, B., de la Rosa, J.L.: *Developing Trust in Recommender Agents*. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS’02). Palazzo Re Enzo Italy, pp. 304–305 (2002)
9. Ohsawa, Y., McBurney, P.: *Chance Discovery, Advanced Information Processing*. Springer, Heidelberg (2003)

10. Pitsilis, G., Marshall, L.: Trust as a Key to Improving Recommendation System. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) *iTrust 2005*. LNCS, vol. 3477, pp. 210–223. Springer, Heidelberg (2005)
11. Pitsilis, G., Marshall, L.: A Proposal for Trust-enabled P2P Recommendation Systems. Technical Report Series (CS-TR-910). University of Newcastle upon Tyne (2005)
12. O'Donovan, J., Smyth, B.: Trust in Recommender Systems. In: *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pp. 167–174 (2005)
13. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pp. 175–186. ACM Press, New York (1994)
14. Resnick, P., Varian, H.: Recommender Systems. *Communications of the ACM* 40(3), 56–58 (1997)
15. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: *Proceedings International WWW Conference*, pp. 285–295 (2001)
16. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating 'Word of Mouth'. In: *Proceedings of the Conference on Human Factors in Computing Systems (CHI95)*, pp. 210–217 (1995)
17. Watts, D.J.: *Six Degrees: The Science of a Connected Age*. W.W. Norton & Company, New York (2003)