A Group Recommendation Approach for Service Selection

Tao Liu
State Key Laboratory for Novel Software Technology, Nanjing University, China
ltao@smail.nju.edu.cn

Feng Xu*
State Key Laboratory for Novel Software Technology, Nanjing University, China
xf@nju.edu.cn

Yuan Yao
State Key Laboratory for Novel Software Technology, Nanjing University, China
yyao@smail.nju.edu.cn

Jian Lu
State Key Laboratory for Novel Software Technology, Nanjing University, China
lj@nju.edu.cn

ABSTRACT

There are more and more services that fulfill similar functionality, such as image service provided by Flickr, Picasa and Facebook. Which should be adopted to construct our software system in the open, dynamic and non-deterministic Internet environment is a key problem. Earlier work[15, 9] analyze this problem from the point view of QoS and established generic and extensible QoS computation framework for service selection. However those framework are almost designed for individuals. As social network emerges and gets widespread, people tend to be more connected and self-organize themselves into groups. Benefits of all members should be considered when we select service for group. In this article, we propose a revised group recommendation algorithm which takes advantage of collaborative filtering technology for service selection. As the experiment demonstrates, our algorithm exhibits high accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*

General Terms

Algorithms, Experimentation

Keywords

Service Selection, Recommender System, Multi-criteria, Group, Collaborative Filtering

1. INTRODUCTION

Increasing services with similar functionalities can now be found on the Internet, making service selection among the alternatives a key issue. For example, when constructing an Internetware application [11, 10], personalized users might need to select the best fit services according to their own preferences over the universal QoS attributes. To this end, *recommender systems* have been widely studied in academia and used in industry to provide customized recommendations for individuals. For example, Amazon's recommender system would suggest books that the user probably like, and Netflix would recommend movies to the user based on his/her previous ratings. Up to date, recommender systems can be mainly classified into three categories: *content-based method*, *collaborative filtering*, and *hybrid method* (See Section 2 for a review).

On the other hand, with the prosperity of Web2.0, Internet has witnessed a shift where people become active in interacting with each other and self-organize themselves into groups. This shift improves user experiences and brings many additional benefits. For example, users can provide their feedbacks about the services that they have interacted with, and rate others about their recommending credibility in Internetware [24]. On the other hand, this shift also brings new challenges to existing recommender systems. One of the challenge is to recommend suitable services for a group of users instead of an individual user. Consider that users in the same group might have difference preferences, recommending a service that can maximize the satisfaction of all the users in the group becomes a challenging problem.

In literature, there exists two group recommender systems [20, 17]. However, these group recommendation algorithms do not consider the multi-criteria QoS issue. That is, services are now not only evaluated by their functionality, but also by some other criteria such as price, execution duration, availability, and reliability.

In this paper, we propose a group recommendation approach to select the suitable services for a group of users based on multi-criteria. In contrast to existing group recommender systems, we incorporate multi-criteria into our group recommendation algorithm. In general, our method consists of the following three steps: (1) *Member profile acquisition* where each member is characterized by a vector representing his/her rating pattern; (2) *Group profile acqui-*

*Corresponding Author.

Internetware'12, October 30–31, 2012, Qingdao, China.
Copyright 2012 ACM 978-1-4503-1462-6 /12/10 ...$15.00.

sition where we generate a virtual user that represents the group by analyzing the characteristics of all members; and (3)*Recommendation* where we calculate similarity between the virtual user and other users and then make the final recommendation. Experimental results show that our method outperforms baseline methods with respect to overall satisfaction.

The rest of this paper is organized as follows. Section 2 covers related work. Section 3 presents our three phase methodological framework. Section 4 presents the experimental results. Finally, we conclude this article and discuss some future work in Section 5.

2. RELATED WORK

Ma et al. [12] propose a requirement driven evaluation framework for Internetware-based service with respect to both functionality and risk. Liu et al. [9] discuss a dynamic QoS computation model for web service selection through the implementation of a QoS registry in a hypothetical phone service provisioning market place. Based on the idea of trust evaluation, Wang et al. [22] propose an Internetware software architecture oriented trust-driven mechanism for selecting services. These methods are all trust based methods which help the users to evaluate the services. In contrast, we focus on how to recommend new services to users.

As for recommender system, it becomes a popular research field today. There are three main classifications for recommender system. The first is content-based method [4] which recommends stuff of the similar property with items that user once bought and gave high ratings. A famous example is the Internet radio service Pandora.com, known as the Music Genome Project. This kind of method is easy to implement but domain-related. The second class is collaborative filtering method [19]. Collaborative filtering weights similarity between items by comparing existing ratings and predicting user's interests towards unrated item. Domain free is one of the major advantages which makes it suitable for situations where items are difficult to profile. The third class is the hybrid method [2] which aims to combine the above two kinds of methods. Nowadays hidden factor method becomes popular, such as matrix factorization technique [7] which won the Netflix prize. Our algorithm belongs to collaborative filtering.

There exist several pieces of work that consider multi-criteria for recommendation. Techniques from decision theory, more specifically the field of Multiple Criteria Decision Making (MCDM) can help to build a model representing the user's preferences. Recently, significant research has been undertaken towards this aspect, Liu et al. [8] assume that some criteria will dominate the overall ranking. In their settings, users are clustered in "preference lattices" and those in the same lattice or nearby lattices are given more weight when collaborative filtering is applied. However, their method needs to run random walk several times which is time-consuming. Park et al. [17] draw a framework and divide the whole recommendation process into four parts including context-log collection, preference modelling, integration, and final recommendation. They use Bayesian network and the Analytic Hierarchy Process [18] as the tool. However, their pair-wise comparison matrix is difficult to acquire. Tsoukias et al. [21] use the UTA* [5] algorithm to process multi-criteria data matrix. again they do not aim for group recommendation.

As for group recommendation, there are two main strategies proposed in literature. The first strategy brings the idea of creating a joint virtual user profile on behalf of the whole group, and making recommendations by applying traditional individual methods to this virtual user. An intuitive way is using shared history [25] to profile the group, but it requires a long initialization period. In our method, we give a simple model to get rid of this problem. The second strategy merges the recommendation lists of all members into one list. For example, Seko wt al. [20] calculate recommendation scores using a feature space that consists of the behavioral tendency of a group and the power balance among group members. However, their method cannot deal with the group with more than three members. Baltrunas et al. [1] compare five different rank list aggregation methods and find that they behave almost equally. Meanwhile, Yu et al. [25] and Berkovsky et al. [3] verify that the approach of merging individual profiles is superior to the one that merges recommendation lists. Merging member's appetite and finally deriving a result satisfied by all is hard, Masthoff et al. [14] conclude that disagreeable experiences are perceived as outweighing possible pleasurable ones. As a result, Our method belongs to the first category.

3. METHODOLOGICAL FRAMEWORK

In this section, we discuss the overall framework of our approach. We analyze rating pattern of every user, combine these properties of all members in the group, and take advantage of collaborative filtering to do recommendation for the group. The process consists of three phases.

3.1 First phase: member profile acquisition

During the data acquisition process, we obtain data of format $User \times Item \rightarrow R_0 \times R_1 \times \ldots \times R_c$, where R_i is User's rating towards item on criterion i and R_0 is the overall score. We realize that people are characterized by their rating pattern, which is the latent factor hidden in their rating data. Instead of asking the users to specify their bias for each criterion, we assume the overall score is a linear combination of rating for every criterion, namely,

$$r_0 = \sum_{i=1}^{c}(w_i * r_i) + \theta \qquad (1)$$

where w_i is weight of criterion i in the overall score, r_i is rating for criterion i and c is the number of criteria. Using multi-criteria linear regression method, we capture every user's weight vector $\vec{w}$ which characterizes him and is later used for similarity calculation. It only makes sense that these coefficients w_i are positive, so in the experiment we filter out those invalid records whose coefficient is negative. To tackle with outlier in dataset, we resort to RANSAC [6], which is an iterative method to estimate parameters of a specified model from a set of observed data containing noise.

3.2 Second phase: group profile acquisition

In this step, we need to create a virtual user to delegate the group. Members of a group may have similar tastes. For example, friends gathered together because of common interests. But most of the time, people in a group may have different preferences towards items. It is hard to analyze the appetite of a group as mentioned before. To take everyone's opinion into account and make the result as consistent as possible, we measure the disagreement of the group, depicted as follows:

$$dis_i = \sum_{j!=i}^{n} ||\vec{w_i} - \vec{w_j}|| \tag{2}$$

$$con_i = \frac{1}{dis_i} \tag{3}$$

$$\vec{w} = \sum_{i=1}^{n} (con_i * \vec{w_i}) \tag{4}$$

where n is the number of members in this group, dis_i stands for member i's disagreement with others, and con_i is his contribution to the final weight vector of the virtual user profile. More sophisticated aggregation models that consider members' inter relationship can be established by means of adjusting their contribution weight. For example, a subset of the group may be more dominant to the final decision. It can be accomplished by user configuration or by learning from users' feedback, and we leave this for future work. Next, we use this generated profile to search his neighborhood.

Under multi-criteria circumstance, there are two kinds of methods to calculate similarity between users. The first is calculating Pearson Correlation for each criterion and then averaging them:

$$sim_j\left(u,v\right) = \frac{\sum_{i \in I} (r_{u,i,j} - \bar{r}_{u,j})(r_{v,i,j} - \bar{r}_{v,j})}{\sqrt{\sum_{i \in I} (r_{u,i,j} - \bar{r}_{u,j})^2} \sqrt{\sum_{i \in I} (r_{v,i,j} - \bar{r}_{v,j})^2}} \tag{5}$$

$$sim\left(u,v\right) = \frac{1}{c} * \sum_{j=1}^{c} sim_j\left(u,v\right) \tag{6}$$

Where I represents the common set of items rated both by user u and user v, $r_{u,i,j}$ stands for rating user u gives to item i on criterion j and $\bar{r}_{u,j}$ is the average score for j. The value of $sim_j\left(u,v\right)$ ranges from -1 to +1. The greater value shows the more similar these two users are, while -1 means they have exact opposite tastes.

The second method is:

$$d_i\left(u,v\right) = \sqrt{\sum_{j=1}^{c} (r_{u,i,j} - r_{v,i,j})^2} \tag{7}$$

$$sim\left(u,v\right) = \frac{1}{1 + \frac{1}{|I|} * \sum_{i \in I} d_i\left(a,b\right)} \tag{8}$$

where $d_i\left(u,v\right)$ represents the difference of interest between user u and v towards item i. It measures the difference using Euclidean Distance.

Based on our experiments, we find that these two methods behave equally. As a result, we will only concentrate on the latter one from now on.

3.3 Third phase: recommendation

The basic idea of CF-based algorithm is to provide item recommendation or prediction based on the opinions of other like-minded users. So at last, we use user-based collaborative filtering to predict virtual user's rating towards unrated items. The final rating is depicted as:

$$r\left(u,i\right) = \frac{1}{\sum_{v \in U} sim\left(u,v\right)} * \sum \left(sim\left(u,v\right) * r\left(v,i\right)\right) \tag{9}$$

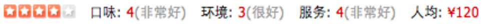
where U is the set of users whose similarity with u exceeds 0. Finally, we sort all items by their estimated ratings, and recommend top items to the group. In case of items in top positions stay unchanged (due to lack of user feedback etc), we could add some random factors, such as those proposed in [23, 16].

In the situation where user v's rating towards target item i is unknown, we would further adopt user v's existing rating for item j which is the most similar to item i. That is to say the value can be acquired by item-based collaborative filtering to handle data sparsity.

4. EXPERIMENT AND ANALYSIS

4.1 Experiment Setup

There is no existing dataset about users' rating towards service. Instead, we choose the situation that resembles. We wrote a script to crawl ratings on www.dianping.com, which is the most extraordinary website of reviews about resturants in China. We collected 30029 comments involving 482 persons and 957 restaurants in total.



★★★★☆ 口味: 4(非常好) 环境: 3(很好) 服务: 4(非常好) 人均: ¥120

Figure 1: A rating from www.dianping.com

As shown in figure 1, a comment consists of ratings for the food's flavor, the dining environment, the quality of service, and the overall score. Other stuff like price and note is ignored by our script. The rating value ranges from 1 to 5. We randomly divide the whole dataset into two parts, 60% are used for training while the rest 40% are used for testing.

In the experiment, we evaluate our algorithm in the matter of a standard IR measure called Normalized Discounted Cumulative Gain (nDGG) [13]. Let $t_1, t_2, \ldots, t_n$ be the calculated rank list(from most enjoyable to least) for user u where each t_i is a restaurant in our dataset. Let r_{ut_i} be the true rank of t_i in u's opinion. Then DCG and nDCG at rank k is defined as:

$$DCG_k^u = r_{ut_1} + \sum_{i=2}^{k} \frac{r_{ut_i}}{log_2\left(i\right)} \tag{10}$$

$$nDCG_k^u = \frac{DCG_k^u}{IDCG_k^u} \tag{11}$$

where IDCG is the maximum possible gain value obtained by re-order of the k items, which we calculate using the test dataset. But to compute nDCG we need to know the true user rating for all items which is impossible due to the dataset sparsity. We adopt the same idea from [1]. That is, we compute nDCG on the projection of the recommendation list on the test dataset. For example, imagine that $r = \{5, 9, 1, 2, 7, 6, 20\}$ is a ranked list of recommendations for a group. Moreover, suppose that the test dataset of user u contains five records $\{5, 7, 8, 9, 12\}$. In such case, we would compute nDCG on the restaurant set $\{5, 9, 7\}$.

We implemented our algorithm using Java and ran the experiment in an Intel Core2 Duo 2.2GHz CPU with 2GB RAM and XP system machine.

4.2 Result and Analysis

The first serial of experiment we performed is to verify the assumption that the overall score is a linear combination of rating for each criterion. After we achieve user's weight vector for different criterion, we apply it to the test data to verify the rationality of linear model. Below are two figures describe the mean of relative error and variance, respectively. The x-axis stands for user id and y-axis is error value. As figure 2 and figure 3 show, the majority of errors is between 0.1-0.2, with a low variance. We consider the pulse as outliers in test dataset.

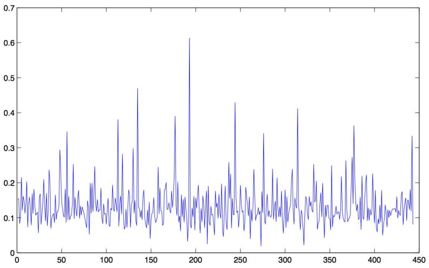


Figure 2: Average of error

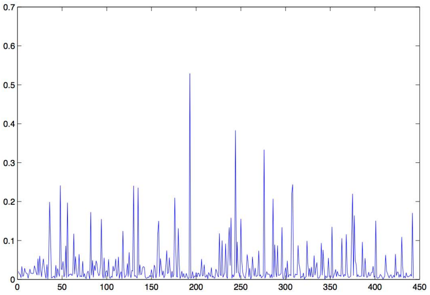


Figure 3: Variance of error

Second, we do experiment on groups of various size to see how size affect the accuracy. We apply our method to groups of size 2 to 7. We randomly choose group member and generate 40 groups in each case. Figure 4 presents the average nDCG with respect to the group size. As we can see, the value is between 0.9 and 0.92. It validates that our method is scalable which could be inferred from formula (2).

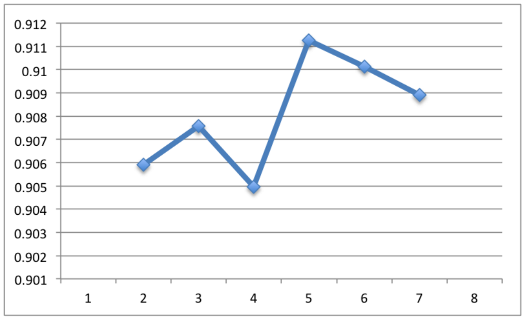


Figure 4: nDCG of different group size

Finally, we compare our method with list merging algorithm which is often used in group recommendation, i.e., the "Average" and "Least Misery" methods mentioned in [1]. As we can see from figure 4, our method does not fluctuate with group size. We choose size three for this test case. Figure 5 presents the comparison results between our algorithm and the two list merging methods (i.e. "Average" and "Least Misery"). The x-axis is group id, and y-axis is nDCG value. As the figure shows, our method outperforms the other two. The nDCG values are relatively large because the size of mapped set is small and the realistic overall score is less distinguishable. As a result, order makes the influence less obvious and of the same reason exceeding part is relatively small. But the long run performance is stable.

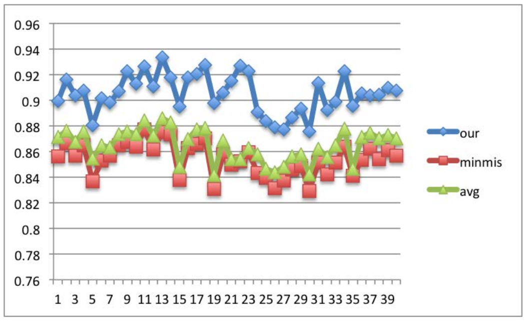


Figure 5: Result comparison

5. CONCLUSION AND FUTURE WORK

In this paper, we solve the service selection problem in Internetware paradigm from a different perspective. Unlike traditional methods that are based on generic QoS and oriented to individuals, we present a novel approach for multicriteria group recommendation. As far as we know, this is the first time to take into consideration both these two restrictions and apply them to service selection. We use linear regression function to discover rating pattern for each group member, create a virtual user to delegate the whole group, and further take advantage of collaborative filtering technique to recommend service. Following these steps, we conduct several experiments and find that the overall satisfaction of our method is higher than the baseline list merging method which is usually used for group recommendation.

One problem about our algorithm is that it suffers from the data sparsity problem, as we need initial data to obtain user's rating pattern. On the other hand, group feedback is ignored here, by considering which we can infer that who plays dominant role in making decisions to remodel our synthesis process accordingly. These are left for future exploration.

6. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 60736015, 61073030), and the National 973 Program of China (2009CB320702).

7. REFERENCES

[1] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and

collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM, 2010.

[2] A. Barragáns-Martínez, E. Costa-Montenegro, J. Burguillo, M. Rey-López, F. Mikic-Fonte, and A. Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.

[3] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 111–118. ACM, 2010.

[4] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In *Proceedings of the 17th international conference on World Wide Web*, pages 1041–1042. ACM, 2008.

[5] J. Figueira, S. Greco, and M. Ehrgott. *Multiple criteria decision analysis: state of the art surveys*, volume 78. Springer Verlag, 2005.

[6] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] L. Liu, N. Mehandjiev, and D. Xu. Multi-criteria service recommendation based on user criteria preferences. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 77–84. ACM, 2011.

[9] Y. Liu, A. Ngu, and L. Zeng. Qos computation and policing in dynamic web service selection. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 66–73. ACM, 2004.

[10] J. Lü, X. Ma, Y. Huang, C. Cao, and F. Xu. Internetware: a shift of software paradigm. In *Proceedings of the First Asia-Pacific Symposium on Internetware*, page 7. ACM, 2009.

[11] J. Lü, X. Ma, X. Tao, F. Xu, and H. Hu. Research and progress on Internetware. *Science in China (Series E)*, 36(10):1037–1080, 2006.

[12] W. Ma, L. Liu, X. Ye, J. Wang, and J. Mylopoulos. Requirements-driven internetware services evaluation. In *Proceedings of the First Asia-Pacific Symposium on Internetware*, page 6. ACM, 2009.

[13] C. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[14] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[15] E. Maximilien and M. Singh. A framework and ontology for dynamic web services selection. *Internet Computing, IEEE*, 8(5):84–93, 2004.

[16] K. Onuma, H. Tong, and C. Faloutsos. Tangent: a novel,'surprise me', recommendation algorithm. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 657–666. ACM, 2009.

[17] M. Park, H. Park, and S. Cho. Restaurant recommendation for group of people in mobile environments using probabilistic multi-criteria decision making. In *Computer-Human Interaction*, pages 114–122. Springer, 2008.

[18] T. Saaty. *Analytic hierarchy process.* Wiley Online Library, 1980.

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[20] S. Seko, T. Yagi, M. Motegi, and S. Muto. Group recommendation using feature space representing behavioral tendency and power balance among members. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 101–108. ACM, 2011.

[21] A. Tsoukiàs, N. Matsatsinis, and K. Lakiotaki. Multi-criteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64, 2011.

[22] Y. Wang, J. Lu, F. Xu, and L. Zhang. Internetware-software-architecture-oriented trust-driven mechanism for selecting services. *Ruan Jian Xue Bao(Journal of Software)*, 19(6):1350–1362, 2008.

[23] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 723–732. ACM, 2010.

[24] Y. Yao, F. Xu, Y. Yang, and L. J. PatTrust: A pattern-based evaluation approach for trust and distrust in internetware. In *Proceedings of the 3rd Asia-Pacific Symposium on Internetware*. ACM, 2011.

[25] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16(1):63–82, 2006.