Dynamic Group Recommendation with Modified Collaborative Filtering and Temporal Factor

Jinpeng Chen, Yu Liu, and Deyi Li
School of Computer Science and Engineering, BeiHang University, China

Abstract: *Group recommendation, which provides a group of users with information items, has become increasingly important in both the workspace and people's social activities. Because users change their preferences or interests over time, the dynamics and diversity of group members is a challenging problem for group recommendation. In this article, we introduce a novel group recommendation method via fusing the modified collaborative filtering methodology with the temporal factor in order to, solve the dynamics problem. Meanwhile, we also put forward a new method of eliminating sparse problem so as to improve the accuracy of recommendation. We have tested our method in the music recommendation domain. Experimental results indicate the proposed group recommender method provides better performance than an original method and gRecs. The result of efficiency and scalability test also shows our method is usable.*

Keywords: *Recommender systems, group recommendation, collaborative filtering, temporal factor, sparsity, dynamics*.

Received August 31, 2013; accepted April 20, 2014; *published online April 1, 2015*

1. Introduction

Recommender system, which attempts to provide some useful information items (news, books, movies, music, etc.,) that the user is expecting to find [1, 10, 11, 12, 13, 18, 37] is one of the most popular research fields. At present, the large majority of recommendation algorithms were designed for individuals [16, 19, 30]. However, there are cases in which many of the items that can be recommended by recommender systems are usually intended for group usage rather than individuals. For example, a group of friends are planning an activity together such as: Finding a restaurant, visiting a tourist attraction, watching a movie, selecting a holiday destination and so on. Therefore, group recommender systems, which suggest the items to a group of people engaged in the activity, have become increasingly important.

Some research works have been carried on to obtain better recommendation results for a group of users [6, 14, 20]. Group recommendation has been applied to different domains such as: News pages, music, tourism and TV program and movies [15, 26, 27, 28, 33, 35]. In the literatures, group recommendation methods are classified into: Aggregating strategy, which merges the user's individual preference recommendation into the recommendation for the whole group; group model, which aggregates user's individual preference model into a group preference model [7].

Most of the studies on group recommendations focus on group formation and evolution, privacy concerns and interfaces for supporting group recommendations. Masthoff researched how humans select a sequence of items for a group to watch, based on data about the individual's preferences, investigated how satisfied people believe they would be with sequences chosen by different strategies, and how their satisfaction corresponds with that predicted by a number of satisfaction functions and explored the influence viewing an item can have on the ratings of other items [25]. Jameson and Smyth [20] selected content with high average scores and low standard deviation to satisfy the group. Boratto *et al.* [8] proposed a group recommendation algorithm, based on user's preferences, that detects communities of similar users and predicts group preferences. Experimental results show that the quality of the recommendations generated by their algorithm improves linearly with the number of communities created. Kim *et al.* [21] they provided a two-step approach: First, recommendations for groups are generated and then items are filtered to increase the individual user satisfaction. Ntoutsi *et al.* [31] they presented an extensive model to exploit recommendations for items in the group recommendation system. They did not exhaustively search for similar users in the whole user base, but they pre-partitioned users into clusters of similar ones and used the cluster members for recommendations. Ntoutsi *et al.* [32] they showed gRecs, a system for group recommendations that follow a collaborative strategy. By partitioning users into clusters of similar ones, recommendations for users are produced with respect to the preference of their cluster members without extensively searching for similar users in the whole user base. In this work, we consider gRecs, which has been shown effective for group recommendations, as a baseline method in order to verify the effectiveness of our proposed method.

However, most existing approaches focus on recommending items of potential interest to a group of users, without taking into consideration how temporal information influences the recommendations. In this paper, we argue that time-aware recommendations need to be pushed in the foreground. We propose a novel group recommendation method that combines the aggregating strategy with the temporal factor. The contributions we make in this paper are as follows:

1. A novel method of eliminating sparsity. We present a novel method of alleviating sparse problem to improve the group recommendation accuracy.
2. A novel strategy measuring the user similarity. Because users change their preferences or interests over time, to capture user's preferences or interests in time, we propose a novel strategy measuring the user similarity by considering a time function which describes user's dynamic behaviours.
3. Group recommendation architecture. We put forward a merging architecture to extend the aggregating strategy utilizing the eliminating sparsity method and the strategy considering the time function for enhancing the group recommendation performance.

The rest of the paper is organized as follows: Section 2 provides an overview of related work, section 3 describes our group recommendation system architecture in detail, section 4 shows our group recommendation strategies and section 5 provides experimental evaluations of the proposed algorithm and compares our algorithm with state-of-the-art methods. Finally, we make some conclusions in section 6.

2. Related Works

A group recommendation system suggests items to a group of people engaged in a group activity. The challenges associated with this simple statement deal with, considering how to record and combine the preferences of many different users as they engage in simultaneous recommendation dialogs. There have been relatively a number of studies on group recommendation systems so far [5, 34].

Gartrell *et al.* [17] studied the key group characteristics that impact group decisions and proposed a group consensus function that captures the social, expertise and interest dissimilarity among group members. Furthermore, they presented a generic framework that can automatically analyze various group characteristics and generate the corresponding group consensus function. Recio-Garcia *et al.* [36] described a group recommender system that takes into account the personality types for the group members. In [38] based on the power balance map and the behavioural tendency of each group, Seko *et al.* [38] proposed an algorithm to recommend appropriate and novel content to groups of people.

However, in the case of group recommender systems only a few have been designed by considering the temporal factor as of now. Backstrom *et al.* [3] examined snapshots of group membership in Live Journal and presented models for the growth of user groups over time. They focused on that the overlaps among pairs of communities change over time. Stefanidis *et al.* [39] studied different semantics to exploit the time information associated with user preferences to improve the accuracy of recommendations. They considered various types of time effects and thus, proposed different time-aware recommendation models. In this work, we also model user's dynamic preferences over time to enhance the accuracy of group recommendations. But our approach is different theirs. To construct user's significant dynamic features, such as: User's mood, user's rating style *et al.*, we show a user cloud similarity measure with temporal factors.

3. The Group Recommendation System Architecture

In this section, we describe the proposed group recommendation architecture in detail.

3.1. Notations Definitions

Assume a set of users $U=\{u_1, u_2, ..., u_n\}$ and a set of items $I=\{i_1, i_2, ..., i_m\}$ (e.g., music, TV programs, books, etc.,) in a recommendation system. Each user u ™U may express a preference for an item i ™I, $Rate_u(u, i)$. Meanwhile, each user u ™U possesses a set of friends or neighbors F_u ™U. For the items unrated by the users, the rating is set to $\perp$. We estimate a predicted rating, denoted as $PreRate(u, i)$, where u ™U and i ™I. To do this, a recommendation strategy with eliminating sparsity is invoked in section 3.2.1.

We further denote a group G, which is made up of two or more people who interact with each other. Here, G ™U and $d=|G|$ is defined as the number of members in G. For instance, if a group is composed of users u_1, u_2, u_3, then it can be defined as $G=\{u_1, u_2, u_3\}$ and $d=|G|=3$. Let $Rate_G(G, i)$ denote the rating for item i given by group G. Let $t=1, 2, ...$, denote a series of times when the items are recommended.

3.2. Influencing Factors

In the proposed group recommendation architecture, three influencing factors are considered: Sparsity, timeliness and dynamics.

Sparsity because the cardinality of the items set I is usually high and typically users rate only a few of these items, the user-item matrix is very sparse. This sparse factor will lead to worse recommendation results. In order to, obtain the recommendation results users are satisfied with, the sparse problem should be addressed in section 3.2.1.

Timeliness nowadays, huge quantities of information emerge every second. An item's popularity may change over time and users also, change their preferences or interests over time. In order to, capture user's preferences or interests in time, we need model user's preferences or interests utilizing a time function in section 3.2.2.

Dynamics with the user changing their preferences or interests, the user similarity also changes over time, which gives rise to the group dynamics. Thus, the method modeling dynamics should be taken into accounts so, as to achieve the precise and novel recommendation results in section 3.2.2.

3.2.1. Eliminating Sparsity

We present the fusion method measuring the user similarity, *SimCC* based on the collaborative filtering methodology and cloud model.

3.2.1.1. Cloud Model

Cloud model is an uncertainty transforming model between a qualitative concept and quantitative numerical values [10, 23].

- ***Definition* 1.** Cloud Model: Let V be a universal set described by a precise number and C be the qualitative related to V. If there is a number x™V, which randomly realizes the concept C and the certainty degree x for C, i.e., $\mu(x)$™ [0, 1] is a random value with stable tendency.

$$\mu : U \rightarrow [0,1], \forall x \in U, x \rightarrow \mu(x)$$

Then, the distribution of x on V is defined as a cloud $C(x)$ and x is defined as a cloud drop. The certainty degree μ also refers as a membership grade.

Cloud model describes a specific concept using Expectation (Ex), Entropy (En) and Hype-entropy (He). *Ex* expresses the point that is the most representative of the qualitative concept and it is the most classical sample while quantifying the concept. *En* represents a granularity of a concept which could be measured (the larger of *En*, the larger of the granularity, the concept is more macro). It reflects the range of the domain space which could be accepted by the specific concept and can be used to express the relationship between randomness and fuzziness. *He* is the uncertain measurement of entropy, i.e., the second-order entropy of the entropy. Vector $v=(Ex, En, He)$ is the eigenvector of a cloud. To get the cloud characteristic vector v of a user, we need a Backward Cloud Generator (BCG) Algorithm 1. The representation of a cloud is shown in Figure 1.

Algorithm 1: *BCG* ($x1, ..., xn$).

Input: *Samples* $x1, ..., xn$
Output: (*Ex*, *En*, *He*) *representing the qualitative concept*

Step 1: *Calculate the mean and variance of* x_i, *i.e.*,

$$\overline{X} = \frac{1}{n} * \sum_{i=1}^{n} x_i$$

$$S^2 = \frac{1}{n-1} * \sum_{i=1}^{n} (x_i - \overline{X})$$

And

$$\overline{\mu_4} = \frac{1}{n-1} * \sum_{i=1}^{n} (x_i - \overline{X})^4$$

Step 2: $Ex = \overline{X}$.

Step 3: $En = \sqrt[4]{\frac{9s^4 - \overline{\mu_4}}{6}}$.

Step 4: $He = \sqrt{S^2 - En^2}$.

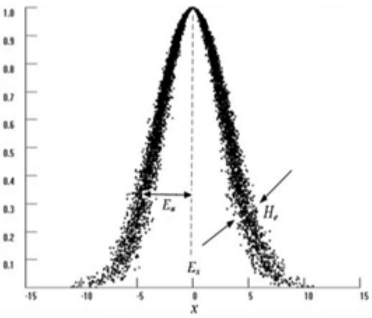


Figure 1. A cloud with Ex=0, En=3, He=0.3, n=10000.

In this paper, the characteristic vector $v=(Ex, En, He)$ of a user cloud, is generated using the BCG and then, the similarity between users is obtained via calculating the cosine-based similarity of v.

- ***Definition* 2.** User Cloud Similarity: Let vector $v=(Ex, En, He)$ denote the cloud of a user u then the similarity $UCSim\ (u_i, u_j)$ between user u_i and u_j is:

$$UCSim\ (u_i, u_j) = cos(v_i, v_j) = \frac{v_i v_j}{\|v_i\| \|v_j\|} \quad (1)$$

Considering that the data set used in this paper is sparse, we choose cosine-based similarity to compute the user cloud similarity. The k most similar users of a target user can be selected by using this method measuring the user similarity.

3.2.1.2. Eliminating Sparsity Method

To avoid the sparsity problem of datasets, we usually estimate a relevance rating of an item for a user. In general, the estimation methods are categorized into: Content-based methods that recommend items similar to those the user has preferred in the past [29], collaborative filtering methods that recommend items that similar user have liked in the past [2] and hybrid methods that combine content-based and collaborative ones [4]. Our work falls into the collaborative filtering category. The collaborative filtering method is only to use preferences of other users that exhibit the most similar behavior to a given user in order to, generate relevance ratings for unrated items, but it omits other significant dynamic features (e.g., user's mood, user's rating style, item's popularity, etc.,). In order to, qualitatively represent these dynamic features, we model them utilizing the cloud characteristic vector v and calculate the vector similarity using the $UCSim(u_i, u_j)$ function.

In this paper, we propose a novel similarity strategy, which is used to locate similar users via fusing the collaborative filtering method with the user cloud similarity.

- ***Definition* 3**. Fusion Strategy: Let α, β>0, α+β=1. The similarity $SimCC(u_i, u_j)$ between user u_i and u_j is:

$$SimCC(u_i, u_j) = \alpha Sim(u_i, u_j) + \beta UCSim(u_i, u_j) \quad (2)$$

Here, $Sim(u_i, u_j)$ is defined using the collaborative filtering method [2]. Although, more sophisticated functions can be designed, the weighted summation of the collaborative filtering similarity and the user's cloud similarity is simple and intuitive. We set α=0.8 and β=0.2 by cross validation. Meanwhile, we note that the collaborative filtering method maps to the case where α=1.0. Thus, to estimate the precise preference of an item recommendation for a user, we propose a predicted rating function leveraging the fusion strategy.

- ***Definition* 4.** Rating Function: The predicted rating of an item *i* ™*I* for a user u_i ™*U* with friends or neighbours F_u is:

$$PreRate(u_i, i) = \frac{\sum_{u_j \in U \text{ and } Rate(u_j, i) \neq \perp} SimCC(u_i, u_j) Rate(u_j, i)}{\sum_{u_j \in U \text{ and } Rate(u_j, i) \neq \perp} SimCC(u_i, u_j)} \quad (3)$$

3.2.2. Timeliness and Dynamics

Time is a kind of important context information, which effect users' interests. Generally, major temporal effects included within the baseline predictors are categorized into: An item's popularity may change over time. For example, movies can go in and out of popularity as triggered by external events such as the appearance of an actor in a new movie. Users may change their baseline ratings over time. For example, a user who tended to rate an average movie "4 stars", may now rate such a movie "3 stars". This may reflect several factors including a natural drift in a user's rating scale, the fact that ratings are given in relationship to other ratings that were given recently and also the fact that the identity of the raters within a household can change over time. In this paper, we construct a temporal influence function based on the latter.

In our method, we take the $f(\Delta t)$ as a function of time delay. Thus, we can model user's dynamic preferences over time by introducing $f(\Delta t)$.

- ***Definition* 5**. Time Delay Function: Let t_{ui} denote the time when a user *u* ™*U* expresses a preference for an item *i* ™*I*. Δt represents the time gap when different users express a preference for the same item. The function of time delay is:

$$f(\Delta t) = \frac{1}{1 + \gamma \Delta t} \quad (4)$$

Here, $\Delta t = |t_{ui} - t_{vj}|$, *u, v* ™*U*. γ is a parameter of time delay, which depends on the recommendations. In other words, the more quickly the user's interest changes, the smaller the value of $f(\Delta t)$ is and when the value of $f(\Delta t)$ is smaller, it indicates the time gap when different users express a preference for the same item is larger.

We use F_u to denote a set of the most similar users for a user *u*. We refer to such users as the friends or neighbors of *u*.

- ***Definition* 6**. Generating Group: In order to, obtain F_u, we need define the dynamic similarity $DSim(u_i, u_j)$ between users over time. The friends or neighbors F_u is formalized using the following similarity:

$$F_u = \begin{cases} Dsim(u_i, u_j) \geq \delta, where \forall u_j \in F_u \\ Dsim(u_i, u_k) < \delta, where \forall u_k \in U / F_u \end{cases} \quad (5)$$

Here, $DSim(u_i, u_j) = SimCC(u_i, u_j) f(\Delta t)$ and δ is a impact factor which controls the intimate degree among F_u.

3.3. Group Recommendation System Architecture

In this section, we describe the main components of our group recommendation system architecture. A high level representation is depicted in Figure 2. Given a set of users, we first generate a group of users via locating the friends or neighbors of each user in the data set. Friend's or neighbour's preferences are employed for estimating personal recommendations, while in turn, personal recommendations are aggregated into recommendations for the whole group using group recommendation strategy. The group recommendation strategy will be introduced in the next section.

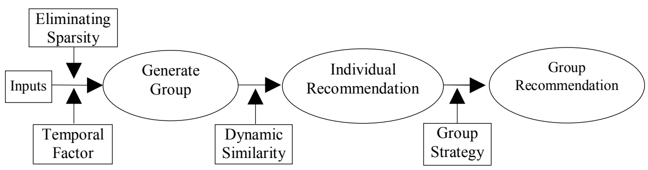


Figure 2. Group recommendation system architecture.

4. Group Recommendation Strategies

In this work, we first compute the personal scores for the unrated items for each user in the group and then, based on these personal scores, we compute the aggregated scores for the group.

- ***Definition* 7.** Group Rating: Given a group of users *G*, *G* ™*U*, the group rating of an item *i* ™*I* for *G* is:

$$Rate_G(G, i) = AggreRate_{u_j \in U \text{ and } Rate(u_j, i) \neq \perp}(GroupRate_{F_u}(u_j, i)) \quad (6)$$

Here, $GroupRate_{Fu}(u_j, i)$ is a specific group recommendation strategy and *Aggregate* is an aggregation function of the group recommendation strategies.

So far, a variety of group decision strategies have been designed. One of the critical goals of the group recommendation is to compute a recommendation score for each item to reflect the preferences or interests of all the group members. Generally, group members can not have the same tastes or ratings for each item. It is important to devise how a group of individuals reach a consensus. In this work, we adopt the three most common group decision strategies [7] including average without misery strategy, most pleasure strategy and least misery strategy.

Average without misery strategy, this strategy assumes the equal importance among all group members and computes the average score of any item rated by the whole group. So, the group rating for item i is computed as follows:

$$GroupRate_{F_u}(u_j, i) = average(Rate_{u_j}(u_j, i)) = \frac{\sum_{j=1}^{d} Rate_{u_j}(u_j, i)}{d} \quad (7)$$

Most pleasure strategy. This strategy supports that a group may choose to rate an item i using the highest rating among all group members. Therefore, items get selected based on their rating on that list, the higher the sooner:

$$GroupRate_{F_u}(u_j, i) = max(Rate_{u_j}(u_j, i)) \quad (8)$$

Least misery strategy, this strategy expresses that the group rating for item i is equal to the smallest predicted rating for among the group members. Formally:

$$GroupRate_{F_u}(u_j, i) = min(Rate_{u_j}(u_j, i)) \quad (9)$$

5. Experiments

5.1. Experimental Setup

In order to, verify the quality of the recommendation, our algorithm was tested using 1K Last.fm dataset [22] which is widely used to evaluate the recommendation algorithm. This dataset represents the full listening history (till May, 5th 2009) for nearly 1,000 users. To evaluate the predicted performance of our algorithm, around 10% of the ratings were extracted as a probe test set and the rest of the dataset was used as a training set for the algorithm.

In the experimental evaluation, we measure the effectiveness of the proposed group recommendation strategies on the baisis of Root Mean Squared Error (RMSE) [8] Appropriate Precision (AP) [9, 38] and normalized Discounted Cumulative Gain (nDCG) [24].

The RMSE can measure the quality of the predicted rating. This metric compares the probe test set with the predicted ratings. *RMSE* is defined by Equation 10:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(PreRate(u,i) - Rate_u(u,i))}{n}} \quad (10)$$

Where n is the number of ratings available in the test set.

AP can measure the accuracy with which the appropriate items are recommended to the subject groups. *AP* is the percentage of the recommended appropriate items from among all recommended items. The formula is shown as follows:

$$AP = \frac{R \cap A}{R} \quad (11)$$

Where R is a set of all recommended items and A is a set of the recommended appropriate items.

nDCG is commonly used in information retrieval to measure the search engine's performance. The higher *nDCG* is the better a ranking results list is. *nDCG* is defined by Equation 12:

$$nDCG@k = \frac{DCG@k}{IDCG@k}, DCG@k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2 i} \quad (12)$$

Where *nDCG@k* measures the relevance of top k results, *IDCG@k* is the *DCG@k* value of ideal ranking list is a relevance value.

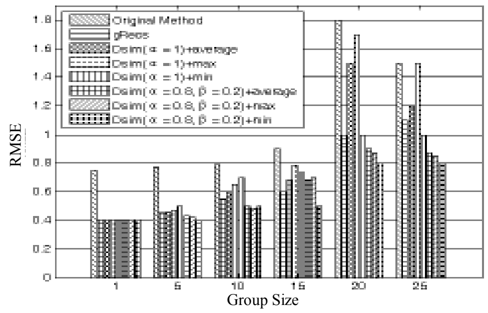
We select two baseline methods: An original method which does not consider the sparse problem and the temporal factor and gRecs which designs the group recommendation strategy via partitioning users into clusters of similar ones [32]. In next section, we will compare these two baselines with our presented group recommendation strategies.

5.2. Experimental Results and Analysis

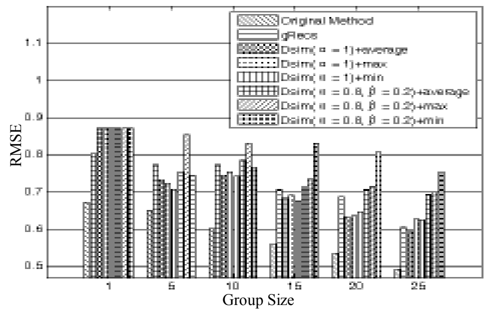
In Figure 3-a, the horizontal axis plots the group size, and the vertical axis plots the value of *RMSE*. From Figure 3-a, we can see that the predictions obtained by our proposed methods are significantly superior to the original method. Computing the average improvement of the best reported strategy (*min* or *max*), we obtain a mean value of 35.7%. Meanwhile, we can observe that the proposed methods (α=1) are closely approximate to gRecs and the proposed methods (α=0.8, β=0.2) outperform gRecs. Computing the average improvement of the best reported strategy (min or max), we obtain a mean value of 11.8%. The possible reason for these is that our proposed group recommendation method not only captures the individual preference of the group members, but also extracts other key factors, such as: Timeliness and dynamics, in the group decision strategy.

In addition, we can notice that, with the query group size d increasing, the prediction accuracy decreases. The explanation for this is that the group recommendations rely on a more diverse set of users and personal values. Here, we find an interesting result: When the group sizes are small and medium, *Dsim* (α=0.8, β=0.2)+*max* method outperforms *Dsim* (α=0.8, β=0.2)+*min* whereas, for the large group sizes, *Dsim* (α=0.8, β=0.2)+*min* is the winner in most of the cases. The performance of *Dsim* (α=0.8, β=0.2)+*average* can be observed to be in between.

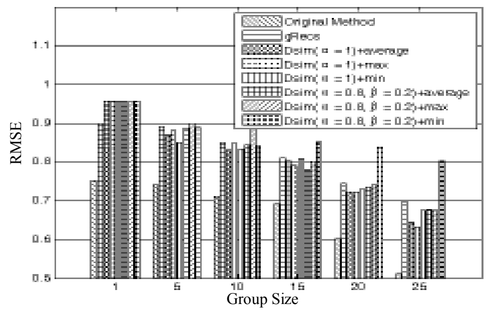
In Figure 3-b, the horizontal axis plots the group size and the vertical axis plots the value of *AP*. We analyzed the behavior of baselines and our proposed algorithms when steadily increasing group size *d*. The results demonstrate that our proposed methods significantly outperform the original method. At the same time, we can notice that the proposed methods (α=1) are closely approximate to gRecs and the proposed methods (α=0.8, β=0.2) exceed gRecs. The explanation for this is similar to ones for Figure 3-a. Because of the space limit, the similar analysis is shown in Figure 3-c.



a) RMSE (δ=0.3).



b) Appropriate precision (δ= 0.5).



c) nDCG@10, δ= 0.7.

Figure 3. Results of the proposed metrics with some state-of-art group recommender algorithms and our presented methods.

6. Conclusions

In this work, a novel efficient framework was presented to recommend items for a group of members by combing the collaborative filtering methodology with the temporal factor. We established that the dynamic similarity function between group members impacts both quality and efficiency and can be exploited to increase the effectiveness of group recommendations. Experimental results show that the proposed method can provide reasonable and high quality group recommendations according to some metrics discussed in section 5.1.

In the future, we intend to extend our work in the following three directions. Firstly, we will attempt to construct the similarity strategy further using more useful features, such as the spatial factor etc. Secondly, we will intend to apply the proposed recommendation method to other services like movies. Finally, we will select more state-of-the-art group recommender algorithms to verify the effectiveness of our presented method.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This work is supported by National Natural Science Foundation of China under Grant No. 61202238, No. 61300006, No. 61035004, No. 61273213, Fundamental Research Funds for the Central Universities under Grant No. YWF-13-D2-XX-07 and the award from the China Scholarship Council.

References

[1] Adomavicius G. and Tuzhilin A., "Context-aware Recommender Systems," available at: http://ids.csom.umn.edu/faculty/gedas/nsfcareer/CARS-chapter-2010.pdf, last visited 2013.

[2] Adomavicius G. and Tuzhilin A., "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions," *IEEE Transaction Knowl Data Engeering*, vol. 17, no. 6, pp. 734-749, 2005.

[3] Backstrom L., Huttenlocher D., Kleinberg J., and Lan X., "Group Formation in Large Social Networks: Membership, Growth and Evolution," *in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Pennsylvania, USA, pp. 44-54, 2006.

[4] Balabanovic M. and Shoham Y., "Fab: Content-based, Collaborative Recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[5] Baltrunas L., Makcinskas T., and Ricci F., "Group Recommendations with Rank Aggregation and Collaborative Filtering," *in Proceedings of the 4th ACM Conference on Recommender Systems*, Barcelona, Spain, pp. 119-126, 2010.

[6] Bento J., Ioannidisy S., Muthukrishnanz S., and Yan J., "Group Recommendations Via Multi-armed Bandits," *in Proceedings of the 21st International Conference on World Wide Web*, Lyon, France, pp. 463-464, 2012.

[7] Berkovsky S. and Freyne J., "Group-based Recipe Recommendations: Analysis of Data Aggregation Strategies," *in Proceedings of the 4th*

ACM Conference on Recommender Systems, Barcelona, Spain, pp. 111-118, 2010.

[8] Boratto L., Carta S., Chessa A., Agelli M., and Clemente L., "Group Recommendation with Automatic Identification of Users Communities," *in Proceedings of the 3rd International Workshop on Distributed Agent-Based Retrieval Tools*, Milan, Italy, pp. 547-550, 2009.

[9] Chen J., Gao H., Wu Z., and Li D., "Tag Co-occurrence Relationship Prediction in Heterogeneous Information Networks," *in Proceedings of International Conference on Parallel and Distributed Systems*, Seoul, Korea, pp. 528-533, 2013.

[10] Chen J., Liu Y., and Zou M., "From Tie Strength to Function: Home Location Estimation in Social Network," *in Proceedings of Conference on Computing, Communications and IT Applications*, Beijing, China, pp. 67-71, 2014.

[11] Chen J., Liu Y., Hu J., He W., and Li D., "A Novel Framework for Improving Recommender Diversity," *in Proceedings of International Workshop on Behavior and Social Informatics*, QLD, Australia, pp. 129-138, 2013.

[12] Chen J., Liu Y., Wu Z., Zou M., and Li D., "Recommending Interesting Landmarks in Photo Sharing Sites," *Neural Network World*, vol. 24, no. 3, pp. 285-308, 2014.

[13] Chen J., Wu Z., Gao H., Zhang C., Cao X., and Li D., *Recommending Interesting Landmarks Based on Geo-tags from Photo Sharing Sites*, Web Information Systems Engineering, Springer, Heidelberg, Berlin, 2013.

[14] Chen Y., Cheng L., and Chuang C., "A Group Recommendation System with Consideration of Interactions among Group Members," *Expert Systems with Applications*, vol. 34, no. 3, pp. 2082-2090, 2008.

[15] Crossen A., Budzik J., and Hammond K., "Flytrap: Intelligent Group Music Recommendation," *in Proceedings of the 7th International Conference on Intelligent User Interfaces*, New York, USA, pp. 184-185, 2002.

[16] Das D. and Horst H., "Recommender Systems for TV," available at: http://aaaipress.org/Papers/Workshops/1998/WS-98-08/WS98-08-008.pdf, last visited 2013.

[17] Gartrell M., Xing X., Lv Q., Beach A., Han R., Mishra S., and Seada K., "Enhancing Group Recommendation By Incorporating Social Relationship Interactions," *in Proceedings of the 16th ACM International Conference on Supporting Group Work*, New York, USA, pp. 97-106, 2010.

[18] Gopalan N. and Batri K., "Effect of Filter Size on Fusion Function in Information Retrieval," *the International Arab Journal for Information Technology*, vol. 5, no. 2, pp. 170-175, 2008.

[19] Isobe T., Fujiwara M., Kaneta H., Morita T., and Uratani N., "Development of a TV Reception Navigation System Personalized with Viewing Habits," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 2, pp. 665-674, 2005.

[20] Jameson A. and Smyth B., *Recommendation to Groups*, the Adaptive Web, Springer, 2007.

[21] Kim H., Kim K., and Ryu Y., "A Group Recommendation System for Online Communities," *the International Journal of Information*, vol. 3, no. 3, pp. 212-219, 2010.

[22] Last.fm Dataset., available at: http://labrosa.ee.columbia.edu/millionsong/lastfm, last visited 2014.

[23] Li D. and Du Y., *Artificial Intelligence with Uncertainty*, National Defense Industry Press, 2005.

[24] Manning D., Raghavan P., and Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[25] Masthoff J., "Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers," *User Modeling and User-Adapted Interaction*, vol. 14, no. 1, pp. 37-85, 2004.

[26] McCarthy J. and Anagnost T., "Musicfx: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts," *in Proceedings of ACM Conference on Computer Supported Cooperative Work*, NY, USA,pp. 363-372, 1998.

[27] McCarthy J., "Pocket Restaurant Finder: A Situated Recommender System for Groups," *in Proceedings of Workshop on Mobile Ad-Hoc Communication at ACM Conference on Human Factors in Computer Systems*, Minneapolis, USA, pp. 1-10, 2002.

[28] McCarthy K., Salam´o M., Coyle L., McGinty L., Smyth B., and Nixon P., "Cats: A Synchronous Approach To Collaborative Group Recommendation," *in Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference*, Florida, USA, pp. 86-91, 2006.

[29] Mooney J. and Roy L., "Content-Based Book Recommending using Learning for Text Categorization," *in Proceedings of ACM DL*, Texas, USA, pp. 195-204, 2000.

[30] Nakamura Y., Itou T., Tezuka H., Ishihara T., and Abe M., "Personalized TV-program Recommendations based on Life Log," *International Conference on Consumer Electronics*, Nevada, USA, pp. 143-144, 2010.

[31] Ntoutsi E., Stefanidis K., Kjetil N., and Kriegel H., "Fast Group Recommendations by Applying User Clustering," available at: https://www.idi.ntnu.no/~noervaag/papers/ER2012.pdf, last visited 2013.

[32] Ntoutsi I., Stefanidis K., Norvag K., and Kriegel P., “Grecs: A Group Recommendation System Based on User Clustering,” available at: https://www.idi.ntnu.no/~noervaag/papers/DASFAA2012.pdf, last visited 2013.

[33] O’Connor M., Cosley D., Konstan J., and Riedl J., “Polylens: A Recommender System For Groups Of Users,” *in Proceedings of the 7th Conference on European Conference on Computer Supported Cooperative Work*, Boston, USA, pp. 199-218, 2001.

[34] Pessemier T., Dooms S., and Martens L., “Design and Evaluation of a Group Recommender System,” *in Proceedings of the 6th ACM Conference on Recommender Systems*, Dublin, Ireland, pp. 225-228, 2012.

[35] Pizzutilo S., De Carolis B., Cozzolongo G., and Ambruoso F., “Group Modeling in a Public Space: Methods, Techniques, Experiences,” *in Proceedings of the 5th International Conference on Applied Informatics and Communications*, Wisconsin, USA, pp. 175-180, 2005.

[36] Recio-Garcia J., Jimenez-Diaz G., Sanchez-Ruiz A., and Diaz-Agudo B., “Personality Aware Recommendations to Groups,” *in Proceedings of the 3rd Conference on Recommender Systems*, New York, USA, pp. 325-328, 2009.

[37] Resnick P., Iacovou N., Suchak M., Bergstorm P., and Riedl J., “Grouplens: An Open Architecture for Collaborative Filtering of Net News,” *in Proceedings of ACM Conference on Computer Supported Cooperative Work*, North Carolina, USA, pp. 175-186, 1994.

[38] Seko S., Yagi T., Motegi M., and Muto S., “Group Recommendation using Feature Space Representing Behavioral Tendency and Power Balance among Members,” *in Proceedings of the 5th ACM Conference on Recommender Systems*, Chicago, USA, pp. 101-108, 2011.

[39] Stefanidis K., Ntoutsi I., Norvag K., and Kriegel P., “A Framework for Time-aware Recommendations,” *in Proceedings of the 23rd International Conference, DEXA 2012*, Vienna, Austria, PP. 329-344, 2012.

Jinpeng Chen is now a PhD candidate in School of Computer Science and Engineering in Beihang University. He received his Master’s degree from Beijingjiaotong University, China, in 2009. His research interests include social network analysis, recommendation system, data mining methodologies, machine learning algorithm and information retrieval technique.

Yu Liu received the PhD degree from Beihang University in 2010. He is now an assistant professor in Department of Computer Science and Technology at Beihang University. His research interests include data mining, cloud computing and natural language processing. He is a member of CCF.

Deyi Li received the BS degree in Electronic Engineering from Southeast University, China, in 1967. He received the PhD degree in Computer Science from Heriot-Watt University, UK, in 1983. He was elected as a member of Chinese Academy of Engineering in 1999 and a member of Eurasian Academy of Science in 2004. He is now a professor in Beihang University and the president of Chinese Association of Artificial intelligence. He has published 4 monographs and over 120 papers on a wide range of topics in artificial intelligence. His current reserach interests include networked data mining, artificial intelligence with uncertainty, cloud computing and cognitive physics.