MLE: A General Multi-Layer Ensemble Framework for Group Recommendation

Xiaopeng Li[1], Jia Xu[1], Bin Xia[1](✉), and Jian Xu[2]

[1] Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, People's Republic of China
bxia@njupt.edu.cn

[2] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
dolphin.xu@njust.edu.cn

Abstract. As the number of users and locations has increased dramatically in location-based social networks, it becomes a big challenge to recommend point-of-interests (POIs) meeting users' preference. In traditional recommendation tasks, personalized recommendations performs well, however, these methods also have many disadvantages such as the long-tailed problem and the strong assumption. Further, in general scenarios, a group of users (e.g., colleagues, friends, and family members) often visit a specific location to enjoy time together (e.g., meal and shopping). Thus, it is more meaningful to recommend locations to the group than to individuals. However, the existing group recommendation approaches also have some limitations that hardly capture the preferences of a group of users effectively. To make full use of the users' preferences and improve the effectiveness of group recommendation, in this paper, we propose a multi-layer ensemble framework which has a two-step fusion process. For the first step, we employ several personalized recommendation methods to generate the recommendations for individuals, and the recommendation list is obtained using the proposed fusion approach based on the supervised learning. For the second step, we utilize several ranking aggregation algorithms to fuse the recommendations list of individuals in the group and propose an unsupervised learning based ranking algorithm (URank) to further fuse the results of ranking aggregations to obtain the final group recommendation list. The experiments are conducted on a real-world dataset, and the results demonstrate the effectiveness of our proposed general framework.

Keywords: Group recommendation · Ranking aggregation · Unsupervised learning · General ensemble model

1 Introduction

In the era of information explosion, recommender systems (RSs) are widely used to address the problem of information overload, where the users can efficiently

© Springer Nature Switzerland AG 2019
X. Liu et al. (Eds.): ICSOC 2018 Workshops, LNCS 11434, pp. 351–366, 2019.
https://doi.org/10.1007/978-3-030-17642-6_29

obtain the valuable information. Recently, the development of smartphones and the location positioning service boosts the emergence of the location-based social networks (LBSNs), such as Foursquare and Gowalla. Using Foursquare APP, users can post check-in records while visiting the locations, and share the experience with friends. These check-in data provide the opportunities to analyze users' preferences and recommend point-of-interests (POIs) for each individual. With the variety of personalized recommendation services, our daily lives become more convenient; hence the location-based recommender system has become one of the hot research topics such as [21,22].

For the personalized recommendations, the major works are divided into many domains, such as: content-based, collaborative filtering, model-based, association rule-based, and hybrid recommendation. The most representative and effective approach is the matrix factorization which is a collaborative filtering based algorithm, such as Biased Matrix Factorization (BMF), Probabilistic Matrix Factorization (PMF), and Neural Collaborative Filtering (NCF), BMF is effective to address the bias of rating from different users, however, it cannot overcome the problem of long-tailed data where the unpopular objects may be hardly recommended. In addition, PMF combines the traditional matrix factorization with the theory of probability and statistics, and predicts users' preferences based on Gaussian distribution. However, the assumption is strong that the data cannot strictly obey the standard normal distribution. Furthermore, NCF, which is a neural network based algorithm, is proposed to solve the problem of the generalized matrix factorization, however, the over-fitting problem affects the performance of recommendation. These state-of-the-art matrix factorization approaches address some significant problems in the personalized recommendation tasks. Meanwhile, each approach has its own disadvantage that decreases the performance of recommendations.

In practical scenarios, the personalized recommendations have some limitations, such as cold-start. If users can be clustered properly, the group recommendations can effectively overcome the problem of missing data and provide fair and appropriate suggestions for group activities. Therefore, ranking aggregation is applied to fuse the personalized recommendation lists to generate group recommendations. Ranking aggregation is mainly categorized into the score-based and order-based aggregation. In the score-based aggregation, each item in a specific recommendation list is given a particular score based on its rank. Fagin's Algorithm (Fagin), which is a classical score-based algorithm, scans the sorted lists in parallel to generate the final aggregation list. However, the effectiveness of the algorithm decreases as the group size increases. Different from the score-based aggregation, the order-based method adopts the order of each item to generate the ranking aggregation list instead of using the score. MedRank has been admitted that is extremely efficient. The rule of MedRank algorithm picks the group's preference that has the best median rank. However, the output of the algorithm is not satisfactory, because the order is used instead of the score for aggregation. Inspired by PageRank, MC_4 algorithm applies the Markov chain to aggregate rankings.

In this paper, we propose a multi-layer ensemble framework which has a two-step fusion process for the group location recommendation. In the personalized recommendation (i.e., the first step), we employ BMF, PMF, and NCF to generate the recommendations, and fuse the recommendations of these models to obtain the personalized recommendation list for each user. In this step, we calculate the weights of each algorithm by the gradient descent method, and re-predict the missing values in the interaction matrix based on these weights. In the group recommendation (i.e., the second step), we propose an unsupervised learning based ranking algorithm (URank), which can be used in the ranking aggregation for the group recommendation and models fusion. Then, we use four ranking aggregation approaches (i.e., Fagin, MedRank, MC_4, and URank) to merge the personalized recommendation lists in the group, respectively. Then, we further fuse the results of these four approaches to obtain the unique recommendation list for the group using URank. In the TopN recommendation problem, experimental results based on the real-world dataset show that our proposed multi-layer ensemble framework has better performance in group recommendation than traditional algorithms.

The main contribution of this paper is summarized below:

1. We propose an unsupervised learning based ranking algorithm (URank). It can be applied not only to the group recommendation task, but also to fuse models.
2. In this paper, we propose a multi-layer ensemble framework which has a two-step fusion process (the personalized recommendation and the group recommendation). In different recommendation processes, we fuse the results of sub-algorithms to remedy disadvantages of them.
3. The experimental results show the effectiveness of our proposed models which outperform other state-of-the-art methods in topN recommendation.

2 Related Work

2.1 Personalized Recommendation

RSs, which play a crucial role to overcome the information overload, have been widely applied in many domains (e.g., news, e-commerce, and social networks). With the development of RSs, a large number of experiments and theories have shown that the matrix factorization (MF) outperforms other algorithms and has become the baseline method to extract the latent feature of users and items. Koren et al. proposed the PMF method, which adopts a probabilistic linear model with Gaussian observation noise to predict the result [8]. It can handle large datasets and deal with users who have few interaction information. Salakhutdinov presented a fully Bayesian treatment of Probabilistic Matrix Factorization (BPMF) by placing hyperpriors over the hyper-parameters to perform approximate inference [18]. Rendle et al. proposed a Bayesian Personalized Ranking (BPR) criterion [17], which is the maximum posterior estimator from a Bayesian analysis and measures the difference between the rankings of user-purchased items and the rest items. With the development of deep neural networks, the algorithm based on neural network has been used in RSs. He et al. proposes a general framework to give the model non-linearities [7]. In other field, Xu et al. designed incentive mechanisms to minimize social costs [23,24].

2.2 Group Recommendation

In the recommendation system, we often encounter such problems: recommend items to a group of users. The method used in most studies is ranking aggregation [4]. Although ranking aggregation is recently being used in a broad range, the origin of it was in the eighteenth century. The two methods popularized at that time was Borda's method and Condorcet method. In many methods based on Borda, Argentini et al. proposed the method which is perhaps the most representative one [1]. It works based on the objects' positions in input rankings directly. Condorcet vote is another popular traditional method, which works based on the pairwise comparisons of items. Dwork et al. used the Locally Kemeny optimal ranking as the fused ranking [3], it is implemented by finding a Hamiltonian path. Moreover, some other methods apply stochastic optimization algorithms such as the cross-entropy Monte Carlo algorithm for searching the optimal ranking [13]. In 2010, Qin et al. proposed a probabilistic model (CPS) [15], which was defined with a coset-permutation distance, and models the generation of a permutation as a stagewise process. In 2015, Ding et al. proposed an instant-runoff ranking fusion method (IRRF) using the result of traditional batch mode ranking fusion methods and a top-2 comparison based instant-runoff ranking fusion method (T2-IRRF), which is an improved IRRF by introducing more local comparison information into the selection of the best item in each round.

2.3 Fusing Recommendation Models

Fusing recommendation models have been well studied to improve the prediction performance; a method combines the predictions of different algorithms, or the same algorithm with different parameters to obtain a final prediction. These algorithms have been successfully used, for instance, in the Netflix Prize contest consisting of the majority of the top performing solutions [14]. The most basic strategy is to acquire the final prediction based on the mean over all the prediction results or the majority votes, such as: Burnham and Anderson [2]. Meanwhile, most of the related works in the literature point out that ensemble learning has been used in recommender system as a way of combining the prediction of multiple algorithms to create a stronger rank, such as linear regression, restricted boltzmann machines (RBM), and gradient boosted decision trees (GBDT), Jahrer et al. propose a fusion model to apply the adaptive learning rate [11], with a modest performance increase. However, the methods mentioned above all have their optimization direction, so the performance improvement of the model is one-sided.

3 Multi-layer Ensemble Framework

3.1 Preliminaries

In this paper, we assume that there exist a set of users $U = \{u_1, u_2, ...\}$ and a set of locations $V = \{v_1, v_2, ...\}$, $\#u$ and $\#v$ denotes the number of users and locations respectively. Therefore, the user-location interaction matrix based on the check-in records is defined as below:

$$y_{ij} = \begin{cases} r_{ij}, \, u_i \; visited \; v_j \\ 0, \quad otherwise \end{cases} \tag{1}$$

where r_{ij} represents the times the user u_i has been to the location v_j. In this paper, the major recommendation algorithms are based on the technique of matrix factorization. Therefore, we employ p_i and q_j to represent the predefined latent features of users and locations, respectively and the preferences of users $\hat{Y}$ (i.e., user-location interaction matrix) are predicted using the latent features.

3.2 Model Framework

Figure 1 illustrates the multi-layer ensemble framework for the group recommendation. The framework is divided into two parts: the personalized recommendation and the group recommendation. For the personalized recommendation part, we consider the interaction matrix of users and locations as the input,

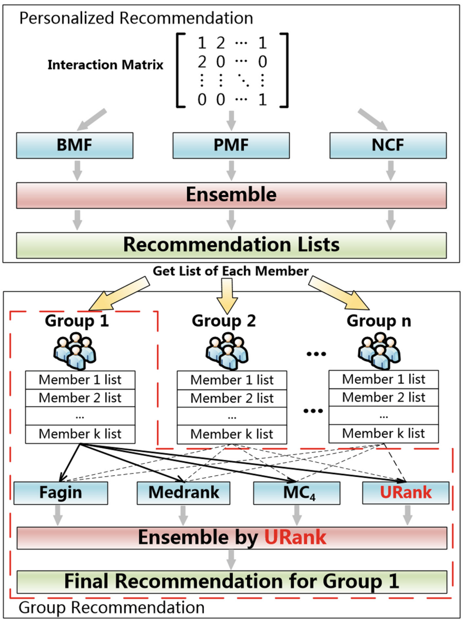

Fig. 1. Model framework

and three model-based recommendation algorithms (i.e., Biased Matrix Factorization, Probabilistic Matrix Factorization, and Neural Collaborative Filtering) are used to generate recommendations for individuals. Then, we propose an ensemble model that fuses the results of the model-based algorithms to obtain a recommendation list for each user. For the second part, we fuse the recommendation list of individuals in the group using four ranking aggregation approaches (i.e., Fagin, MedRank, MC_4, and URank) to obtain the group recommendation lists. Then we employ URank, which is our proposed unsupervised learning based ranking algorithm, to further fuse the results of four ranking aggregations for providing the final recommendation list for each group. Due to the multi-layer ensemble of personalized and group recommendation approaches, the effects of disadvantage of each approach will be decreased and users' preferences can be captured effectively.

3.3 Personalized Recommendation

For the personalized recommendation part, Biased Matrix Factorization (BMF), Probabilistic Matrix Factorization (PMF), and Neural Collaborative Filtering (NCF) are applied to generate the recommendation list for individuals.

BMF is a traditional matrix factorization approach considering the rating bias of each user [8]. In other words, due to the characteristics of individuals, the distribution and preference of rating (or check-in) will be totally different. For a user, 3 (the range of rating is 0–5) may be a general rate, but for another user, 3 may be a high rate. Therefore, the consideration of rating (or check-in) bias will decrease the effect according to the characteristics of individuals. The biased matrix factorization can be defined as follows:

$$b_{ij} = \mu + b_i + b_j, \tag{2}$$

$$\hat{y}_{1ij} = \mu + b_i + b_j + p_i^T q_j, \tag{3}$$

$$min \sum (y_{ij} - \hat{y}_{1ij})^2 + \lambda \|\Theta\|, \tag{4}$$

where b_{ij} represents the rating (or check-in) bias of the user u_i in the location v_j. Then we minimize the error between the actual value y_{ij} and the predicted value $\hat{y}_{1ij}$ using gradient descent to obtain b_i and b_j for predicting the unbiased rating (or check-in).

PMF is a matrix factorization approach considering the perspective of probability [12]. PMF has two assumptions: (1) the prediction error (between the actual value and the predicted value) obeys the Gaussian distribution, and (2) the latent features of users and locations also obey Gaussian distributions. Therefore, the distribution of prediction error and latent features can be defined as follows:

$$p\left(Y|P, Q, \sigma^2\right) = \prod_{i=1}^{\#u} \prod_{i=1}^{\#v} \left[N(y_{ij}|p_i^T q_j, \sigma^2)\right], \tag{5}$$

$$p(P|\sigma_P^2) = \prod_{i=1}^{\#u} N(p_i|0, \sigma_P^2 I), \tag{6}$$

$$p(Q|\sigma_Q^2) = \prod_{j=1}^{\#v} N(q_j|0, \sigma_Q^2 I). \tag{7}$$

Then, the latent features of users and locations will be obtained by minimizing the loss function as below:

$$minE = \frac{1}{2}\sum_{i=1}^{\#u}\sum_{j=1}^{\#v} I_{ij}(y_{ij} - \hat{y}_{2ij}) + \frac{\lambda_P}{2}\left\|P\right\|^2 + \frac{\lambda_Q}{2}\left\|Q\right\|^2, \tag{8}$$

$$\hat{y}_{2ij} = p_i^T q_j, \tag{9}$$

where I_{ij} denotes whether the user u_i has been to the location v_j, and λ_P and λ_Q are regularization parameters.

From the perspective of neural networks, NCF introduces non-linearity to learn the interaction information, rather than a handcraft that has been done by the traditional matrix factorization model [7]. The prediction function of NCF model is defined as follows:

$$\hat{y}_{3ij} = a_{out}\left(h^T\left(p_i \odot q_j\right)\right) \tag{10}$$

where a_{out} and h denotes the activation function and edge weights of the output layers, respectively. Intuitively, if we allow h to be learned from data without the uniform constraint, it will result in a variant of MF that allows varying importance of latent dimensions. If we use a non-linear function for a_{out}, it will generalize MF to a nonlinear setting which might be more expressive than the linear MF model. In this paper, we implement a generalized version of MF under NCF that considers the sigmoid function $\sigma(x) = 1/\left(e^{-x}\right)$ as a_{out} and learns h from data with the log of loss.

BMF, PMF, and NCF have good performance in general scenarios, however, these approaches also have their disadvantages. For example, BMF doesn't work well if the dataset has the long-tailed problem; PMF works based on the strong assumptions; NCF, as a neural network based approach, will overfit the dataset. Therefore, we propose an approach to fuse the results of three basic models $(\hat{Y}_1, \hat{Y}_2, \hat{Y}_3)$ to decrease the effects of these disadvantages. In this paper, we fuse the recommendations lists based on RMSE between the ground truth and the predicted value. The loss function (i.e., RMSE) is defined as below:

$$RMSE = \sqrt{\frac{1}{\#u\#v}} \cdot \sqrt{\sum_{i=1}^{\#u}\sum_{j=1}^{\#v}\left(y_{ij} - \sum_{k=1}^{3} d_k \hat{y}_{kij}\right)^2} \tag{11}$$

where $d_k(k = 1, 2, 3)$ denotes the weight of each basic algorithm. Equation 11 is a continuously derivable function, so the parameter d_i can be deduced using the gradient descent method, and the obtained weights $d_k(k = 1, 2, 3)$ determine the impact of each basic model on the final recommendation list. The gradient direction of d_k is defined as below:

$$\nabla d_k = \frac{\partial RMSE}{\partial d_k} = \frac{1}{2}\sqrt{\frac{1}{\#u\#v}}\left[\sum_{i=1}^{\#u}\sum_{j=1}^{\#v}\left(y_{ij} - \sum_{k=1}^{3} d_k \hat{y}_{kij}\right)^2\right]^{-\frac{1}{2}}$$

$$\cdot \sum_{i=1}^{\#u}\sum_{j=1}^{\#v} 2\left(y_{ij} - \sum_{k=1}^{3} d_k \hat{y}_{kij}\right)(-\hat{y}_{kij}), \tag{12}$$

and the update formula of d_k is:

$$d_k = d_k - \gamma \nabla d_k \tag{13}$$

where γ denotes the learning rate of the method. When Eq. 11 coverages, the final recommendation list for individuals will be obtained based on the ensemble model considering the learned weights $d_k (k = 1, 2, 3)$.

3.4 Group Recommendation

The group recommendation part aims to aggregate recommendation lists of group members into a unique recommendation list for the group. In this part, we employ four ranking aggregation approaches (i.e., MedRank, Fagin, MC_4, and URank) to fuse the recommendation lists. These approaches will be briefly introduced in the following sections.

MedRank is an order-based rank aggregation algorithm [6]. The main idea of MedRank is: the item, which appeared in more than half of the number of ranking lists, will be remained in the final aggregated ranking. For example, given three rankings $R_1 : [A, B, C, D]$, $R_2 : [B, A, D, C]$, $R_3 : [B, C, A, D]$. To count the elements in the first position of each ranking list, we find that the number of B is 2, which is more than half of the number of rankings ($2 > 1.5$). Therefore, B is considered as the first position in the aggregated ranking list. Then, B is removed from the three ranking lists and the elements in the first position will be counted and selected. In this way, we will obtain the aggregated ranking $[B, A, C, D]$.

Different from MedRank, Fagin is a score-based rank aggregation algorithm and is well-suited for the group recommendations [5]. For example, there are three rankings: $R_1 : [A : 1.0, B : 0.8, C : 0.5, D : 0.3, E : 0.1]$, $R_2 : [B : 0.8, C : 0.7, A : 0.3, D : 0.2, E : 0.1]$, $R_3 : [D : 0.8, C : 0.6, A : 0.2, E : 0.1, B : 0]$. If a top-2 recommendation is demanded, then we take the top-2 elements from these rankings into a new list until all values of k elements in the new list are taken in the original three lists, and $C : [0.5, 0.7, 0.6], B : [0.8, 0.8, 0], A : [1.0, 0.3, 0.2], D : [0.3, 0.2, 0.8]$ are the generated. Therefore, the aggregated list is $[C : 0.6, B : 0.53, A : 0.5, D : 0.43]$.

Because the ranking aggregation is similar to traditional page rank, Dwork et al. creatively applies the Markov chain (MC) to aggregate rankings, and proposes MC series algorithm [4]. These algorithms calculate their transition probability matrix by different rules, then get a new sort according to the smooth distribution of the transition matrix. The rule followed by the MC_4 algorithm is as follows: choose the candidate a, then pick another candidate b uniformly from the union of all candidates ranked by vectors. If most voters ranked b higher than a, go to b. Otherwise, stay in a.

URank, which is our proposed unsupervised ranking aggregation method, considers normalized Discounted Cumulative Gain (nDCG) as the ranking metric, specifically as follows: r_{t_j} represents the actual rating of the item t_j (ranked in position j, i,e., $\sigma(t_j) = j$). DCG and nDCG at Top-n are defined as:

$$DCG = r_{t_1} + \sum_{j=2}^{n} \frac{2^{r_{t_j}}}{\log_2(j)} \tag{14}$$

$$nDCG = \frac{DCG}{IDCG} \tag{15}$$

where $IDCG$ is the maximum possible gain value that is obtained with the optimal re-order of the n items in $t_1, ..., t_n$.

If nDCG is considered as the metric to fuse the multiple ranking lists, there exist two significant problem: **a**. nDCG is often applied to evaluate the ordering performance with labels. However, the current scenario is an unsupervised problem, we cannot utilize nDCG according to the traditional point of view. **b**. Because the ranking list is ordered by a series of discrete values, it is non-trivial to learn weights using the gradient descent. To overcome the problem **a**, we propose a strategy instead of the traditional way: (1) initialize the weights of different models; (2) get the integrated order by the weighted results and calculate nDCG between integrated order and the order of each basic model; (3) calculate the accumulation of nDCG. The formula is defined as below:

$$E(w) = \sum_{k=1}^{3} \frac{1}{IDCG_k} \sum_{j=1}^{\#v} \frac{2^{s_{kj}} - 1}{\log_2(1 + \pi(j))}, \tag{16}$$

where $IDCG_k$ denotes the normalization parameter and k denotes the index of the models. $\pi(j)$ is a symbolic function and denotes the ranking position of the location v_j:

$$\pi(j) = 1 + \sum_{m=1, m \neq j}^{\#v} I\left[f_w(j) \succ f_w(m)\right], \tag{17}$$

where $\succ$ denotes the order relationship of the rank, and $f_w(j)$ is a linear function that represents the sort score of the location v_i. I is a 0–1 recognition function:

$$I = \begin{cases} 1, & f_w(j) \succ f_w(m) \\ 0, & other \end{cases} \tag{18}$$

Therefore, Eq. 16 can be rewritten as:

$$E(w) = \sum_{k=1}^{3} \frac{1}{IDCG_k} \sum_{j=1}^{\#v} \frac{2^{s_{kj}} - 1}{\log_2\left(2 + \sum_{m=1, m \neq j}^{\#v} I\left[f_w(j) \succ f_w(m)\right]\right)} \tag{19}$$

Note that, Eq. 19 is not a continuous function, so we cannot calculate the optimal w by partial derivatives, this is the problem **b** mentioned above. To overcome the problem **b**, a natural way for the approximation is to approximate the indicator function I using a logistic function [16]. Therefore, $\pi(j)$ can be replaced using $\hat{\pi}(j)$:

$$\hat{\pi}(j) = 1 + \sum_{m=1;m\neq j}^{\#v} \frac{exp(-\alpha(f_w(j) - f_w(m)))}{1 + exp(-\alpha(f_w(j) - f_w(m))} \tag{20}$$

Table 1 shows the comparison between $\hat{\pi}(j)$ and $\pi(j)$. When $\alpha > 0$ is a scaling constant(e.g., 50, 100, 150), $\hat{\pi}(j)$ is a continuous and differentiable function. Thus, Eq. 19 can be rewritten as:

Table 1. Examples of position approximation

object	s_i	$\pi(i)$	$\hat{\pi}(x)(\alpha = 100)$
$object_1$	4.20074	2	2.00118
$object_2$	3.12378	4	4.00000
$object_3$	4.40918	1	1.00000
$object_4$	1.55258	5	5.00000
$object_5$	4.13330	3	2.99882

$$E(w) = \sum_{k=1}^{3} \frac{1}{IDCG_k} \sum_{j=1}^{\#v} \frac{2^{s_{kj}} - 1}{\log_2\left(2 + \sum_{m=1;m\neq j}^{\#v} \frac{exp(-\alpha(f_w(j) - f_w(m)))}{1+exp(-\alpha(f_w(j) - f_w(m))}\right)} \tag{21}$$

It can be proved that Eq. 21 is a convex function. To solve Eq. 21, we use the gradient ascent based on the chain derivation rule:

$$\nabla w_k = \frac{\partial E(w)}{\partial w} = \sum_{k=1}^{3} \frac{1}{IDCG_k} \sum_{j=1}^{\#v} \frac{\frac{2^{s_{kj}}-1}{\log_2(1+\hat{\pi}(j))}}{\partial \hat{\pi}(j)} \frac{\hat{\pi}(j)}{\partial w ß_k} \tag{22}$$

The updated w_k is defined as below:

$$w_k = w_k + \eta \nabla w_k \tag{23}$$

Therefore, in this paper, we use the aforementioned approaches (i.e., MedRank, Fagin, MC_4, and URank) to fuse the recommendation lists of group members to generate a recommendation list for the group respectively. Then URank is applied to fuse the results from each ranking aggregation approach to obtain the unique ranks for the group.

4 Experimental Evaluation

4.1 Experimental Setup

Check-In Dataset. We evaluated the performance of our method on LBSN datasets. Foursquare is a famous LBSN that allows people to post check-in records in Twitter when they visit a specific venue based on the location information. We collected user's check-in data by calling Twitter and Foursquare's API interface. The dataset contains 419,509 tweets published by 49,823 users among 18,899 locations from August 2012 to July 2013 in Manhattan [19].

Baseline. To evaluate the effectiveness of our proposed model, in the personalized recommendation part, we consider the traditional recommendation algorithms (i.e., BMF, PMF, and NCF) as the baseline method, respectively. In the group recommendation part, we use three ranking aggregation algorithms (i.e., Fagin, MedRank, and MC_4) and the proposed algorithm (i.e., URank) to generate the final group recommendation results based on the personalized recommendations using BMF, PMF, and NCF, respectively. Therefore, the combination of a personalized recommendation algorithm and a ranking aggregation algorithm is considered as a baseline model. Based on different combinations of personalized recommendation algorithms and ranking aggregations, we conducted 20 models on the real-world dataset.

Evaluation Metrics. To assess our proposed framework, we use well-known criteria for TopN recommendation, such as Recall, Precision, and F1-score. The criteria are defined as below:

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}$$

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall}$$

where U is the set of all users, $R(u)$ is the recommendation list based on the preference of user u in the training dataset, and $T(u)$ is the set of user u's behaviors in the test dataset.

4.2 Results and Discussion

In this section, we aim to evaluate the effectiveness of the multi-layer fusion framework for the group location recommendation and answer the following questions:

- Which is the best combination for the recommendation framework?
- What is the performance of proposed framework on different types of groups?

Effectiveness. In this section, we compare the 20 combinations of personalized and group recommendation approaches, and the experimental results are shown in the Table 2. We analyze the experimental results from two aspects: personalized recommendations and group recommendations. For personalized recommendation, we compare the three traditional recommendation algorithms (i.e., BMF, PMF, NCF) with the fusion algorithm we proposed. As observed in Table 2, our proposed fusion model outperforms other traditional personalized recommendation approaches in RMSE. It means that the model we proposed has more strong ability to fit data than the other three models. In addition, the performance of our framework at TopN (N=20, 40, 60) location recommendations for the group also outperforms other models. We also compare the ensemble model with each ranking aggregation (i.e., Fagin, MedRank, and MC_4). From the Table 2, we can see that the proposed framework has better performance in the group recommendation than each ranking aggregation algorithm. The experimental results show that the multi-layer ensemble framework effectively decrease the effect of disadvantages in each approach.

Table 2. Performance comparison

Method	Person	$Fuse_2$	N=20			N=40			N=60		
	RMSE	nDCG	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
BMF-Fagin	0.897	null	0.086	0.088	0.087	0.079	0.135	0.097	0.072	0.182	0.103
BMF-MedRank	0.897	null	0.087	0.091	0.089	0.072	0.131	0.093	0.071	0.180	0.102
BMF-MC_4	0.897	null	0.082	0.096	0.091	0.078	0.133	0.098	0.073	0.181	0.104
BMF-$URank$	0.897	null	0.089	0.087	0.088	0.082	0.140	0.103	0.074	0.181	0.105
BMF-$fuse_2$	0.897	0.891	0.090	0.087	0.088	0.080	0.137	0.101	0.076	0.179	0.107
PMF-Fagin	0.879	null	0.102	0.081	0.090	0.091	0.130	0.107	0.082	0.181	0.113
PMF-MedRank	0.879	null	0.098	0.083	0.090	0.093	0.128	0.108	0.083	0.187	0.115
PMF-MC_4	0.879	null	0.091	0.079	0.085	0.083	0.121	0.098	0.077	0.166	0.105
PMF-$URank$	0.879	null	0.099	0.086	0.092	0.096	0.131	0.111	0.085	0.179	0.115
PMF-$fuse_2$	0.879	0.887	0.103	0.085	0.093	0.101	0.131	0.114	0.084	0.183	0.115
NCF-Fagin	0.872	null	0.108	0.075	0.089	0.092	0.123	0.105	0.083	0.166	0.111
NCF-MedRank	0.872	null	0.103	0.073	0.085	0.093	0.127	0.107	0.085	0.172	0.114
NCF-MC_4	0.872	null	0.096	0.071	0.082	0.082	0.109	0.094	0.075	0.154	0.101
NCF-$URank$	0.872	null	0.107	0.073	0.087	0.090	0.121	0.103	0.082	0.174	0.111
NCF-$fuse_2$	0.872	0.902	0.111	0.074	0.089	0.104	0.129	0.115	0.086	0.170	0.114
$fuse_1$-Fagin	0.853	null	0.120	0.082	0.097	0.104	0.142	0.120	0.093	0.192	0.126
$fuse_1$-MedRank	0.853	null	0.118	0.080	0.096	0.105	0.143	0.121	0.096	0.195	0.128
$fuse_1$-MC_4	0.853	null	0.110	0.075	0.089	0.088	0.120	0.102	0.083	0.169	0.111
$fuse_1$-$URank$	0.853	null	0.122	0.081	0.097	0.103	0.142	0.119	0.089	0.187	0.121
$fuse_1$-$fuse_2$	0.853	0.893	**0.125**	**0.085**	**0.101**	**0.103**	**0.147**	**0.121**	**0.103**	**0.208**	**0.138**

Different Combination. To evaluate the combination of personalized recommendation approaches and ranking aggregations for the group recommendation, we divide the experiments into two categories: re-merge with unused approach and re-merge with used approach. In the first category, we select three of the ranking aggregation algorithms (i.e., Fagin, MedRank, MC_4, URank) as sub-algorithms, and the remaining one is used as the method of fusing models to

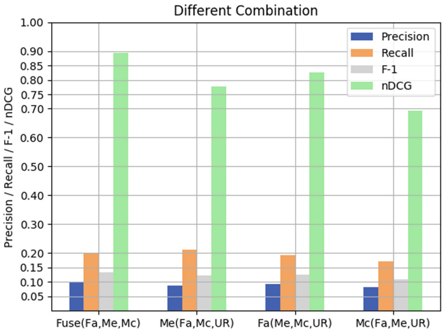

Fig. 2. Triple combination.

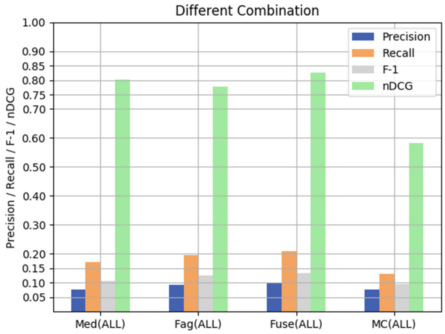

Fig. 3. Quadruple combination.

generate the final result. As observed in Fig. 2, we can find that the performance of the group recommendation is worse than other combinations significantly when the MC_4 algorithm is applied to re-merge the results of models.

In addition, if URank is used to re-merge the recommendation lists aggregated by Fagin, MedRank, and MC_4, the recommendations have better performance. In the second category, we use all four ranking aggregation algorithms as sub-algorithms, and apply an algorithm from them as the method to re-merge the recommendation lists of models (See Fig. 3). As observed in Fig. 3, if we re-merge the recommendation lists of four ranking aggregations using URank, the performance is better than that of other combinations. Note that, compared the experimental results in Figs. 2 and 3, the overall performance will be decreased if we re-merge the recommendation lists with used approaches.

Group Method. To evaluate the performance of proposed framework on different types of groups, we use two ways to group users and conduct the experiments respectively. (I) Random grouping: divide all users into groups randomly; (II) Similar grouping: group similar users based on the interaction matrix.

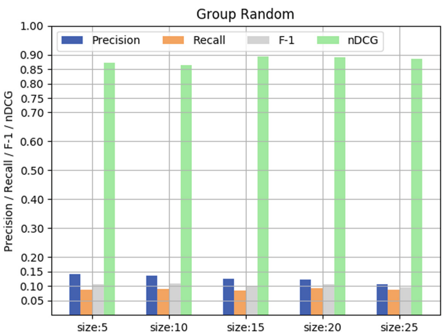

Fig. 4. Group random

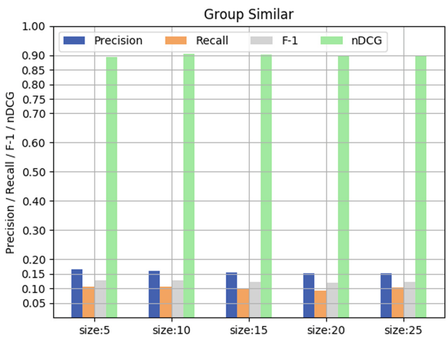

Fig. 5. Group similarity

As observed in Figs. 4 and 5, our proposed framework has better performance in the similar group at the small group size. In addition, compared with Precision, Recall, F1-score of similar groups and random groups, the performance of similar groups is better than that of random groups. This is because that the personalized recommendation lists in the similar group have many intersections. Therefore, when performing TopN group recommendation, the ranking aggregation algorithm tends to select the same object meet members' preference.

5 Conclusion and Future

In this paper, we propose the general multi-layer ensemble framework for the group recommendation. We evaluate the framework on a real-world dataset and the experimental results show the effectiveness of our approach. In addition, we further evaluate the performance of the proposed framework in two aspects: (1) different combinations of algorithms and (2) the effect of the grouping method. For the first question, in the group recommendation part, the experimental results show that if URank is used to further fuse the results of four ranking aggregation methods, the two-step fusion will improve the performance of recommendations. For the second question, the performance of ensemble framework also works better in the similar groups than the random groups.

In the future, we want to continue our research and explore the relationship between group recommendation and personal recommendation; we also hope to validate our findings through experiments with other datasets. Since the results of personalized recommendations have a significant impact on the results of group recommendations, we will try different personalized recommendation algorithms and ranking aggregation algorithms to further improve our performance. Meanwhile, according to life experience, the decision of the group is affected by each member is different in some cases, so weighting the results of the members in the group is also necessary for research. We also hope introduce methods in other fields(event mining) to perfect our framework [9,10,20].

Acknowledgments. The work was supported in part by the National Natural Science Foundation of China (Grant No. 61472193, No. 61872193, No. 61802205 and No. 61872186), the Natural Science Research Project of Jiangsu Province under Grant 18KJB520037, and the research funds of NJUPT under Grant NY218116.

References

1. Argentini, A.: Ranking aggregation based on belief function theory. Ph.D. thesis, University of Trento (2012)
2. Burnham, K.P., Anderson, D.R.: Model selection and multimodel inference: a practical information-theoretic approach. Springer, New York (2003). https://doi.org/10.1007/b97636
3. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation revisited (2001)

4. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th International Conference on World Wide Web, pp. 613–622. ACM (2001)
5. Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 301–312. ACM (2003)
6. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. J. Comput. Syst. Sci. **66**(4), 614–656 (2003)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. International World Wide Web Conferences Steering Committee (2017)
8. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
9. Liu, Z., Li, T., Wang, J.: A survey on event mining for ict network infrastructure management. ZTE Commun. **14**(2), 47–55 (2016)
10. Liu, Z., Li, T., Zhou, Q.: Application driven big data mining. In: ZTE Technology, pp. 49–52 (2016)
11. McKenzie, T.G., et al.: Novel models and ensemble techniques to discriminate favorite items from unrated ones for personalized music recommendation. In: Proceedings of the 2011 International Conference on KDD Cup 2011, vol. 18, pp. 101–135. JMLR.org (2011)
12. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
13. Pihur, V., Datta, S., Datta, S.: Rankaggreg, an R package for weighted rank aggregation. BMC Bioinform. **10**(1), 62 (2009)
14. Piotte, M., Chabbert, M.: The pragmatic theory solution to the Netflix grand prize. Netflix Prize Documentation (2009)
15. Qin, T., Geng, X., Liu, T.Y.: A new probabilistic model for rank aggregation. In: Advances in Neural Information Processing Systems, pp. 1948–1956 (2010)
16. Qin, T., Liu, T.Y., Li, H.: A general approximation framework for direct optimization of information retrieval measures. Inf. Retrieval **13**(4), 375–397 (2010)
17. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
18. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain Monte Carlo. In: Proceedings of the 25th International Conference on Machine Learning, pp. 880–887. ACM (2008)
19. Xia, B., Li, T., Li, Q., Zhang, H.: Noise-tolerance matrix completion for location recommendation. Data Mining Knowl. Discov. **32**(1), 1–24 (2018)
20. Xia, B., Li, T., Zhou, Q.F., Li, Q., Zhang, H.: An effective classification-based framework for predicting cloud capacity demand in cloud services. IEEE Trans. Serv. Comput. (2018)
21. Xia, B., Li, Y., Li, Q., Li, T.: Attention-based recurrent neural network for location recommendation. In: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–6. IEEE (2017)
22. Xia, B., Ni, Z., Li, T., Li, Q., Zhou, Q.: Vrer: context-based venue recommendation using embedded space ranking svm in location-based social network. Expert Syst. Appl. **83**, 18–29 (2017)

23. Xu, J., Li, H., Li, Y., Yang, D., Li, T.: Incentivizing the biased requesters: truthful task assignment mechanisms in crowdsourcing. In: 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–9. IEEE (2017)
24. Xu, J., Rao, Z., Xu, L., Yang, D., Li, T.: Mobile crowd sensing via online communities: incentive mechanisms for multiple cooperative tasks. In: IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 171–179. IEEE (2017)