Collaborative Filtering and Rating Aggregation Based on Multicriteria Rating

Hiroki Morise
Hokkaido University
morise.hiroki@complex.ist.hokudai.ac.jp

Satoshi Oyama
Hokkaido University/RIKEN AIP
oyama@ist.hokudai.ac.jp

Masahito Kurihara
Hokkaido University
kurihara@ist.hokudai.ac.jp

Abstract—**Ratings by users on various items such as hotels and movies have become easily available on the Web. In many cases, other than overall rating for each item by each user, more detailed information such as ratings from different viewpoints and free text comments, as well as aggregated information such as the average of ratings by different users, are also available. We investigated the effectiveness of six existing collaborative filtering methods for large-scale sparse multicriteria rating data. We formulated rating aggregation as a collaborative filtering problem and applied six collaborative filtering methods to it. Furthermore, we extended three of the methods to calculate user similarity using indirect users and review comments and applied them to collaborative filtering and rating aggregation. The results show that multicriteria rating approaches perform better than single criterion rating approaches. The extended methods had better performance both in collaborative filtering and in rating aggregation.**

Index Terms—**Recommendation; multi-criteria rating; collaborative filtering; rating aggregation**

I. INTRODUCTION

Advances in information technology have led to ratings by users on various items such as products, hotels, and movies being easily available on the Web. In many cases, in addition to an overall rating for each item by each user, more detailed information such as ratings from different viewpoints and free text comments, as well as aggregated information such as the average of ratings by different users, are also available. As a result, when many consumers buy products, for example, they make decisions by using these ratings. However, it is difficult to acquire reliable information because of information overload—there is a lot of information such as reputation and opinions for each item on the Web. Therefore, techniques for aggregating ratings and providing information matching the user's preferences have been attracting attention. Recommendation techniques in particular help consumers avoid information overload and find interesting items.

One of the most promising types of recommendation methods is collaborative filtering (CF). User-based CF is used to compute similarities among users on the basis of ratings they previously provided and then to predict the unknown rating for an item. As demonstrated by the Netflix Prize competition, matrix factorization, which uses a matrix of users and items, is superior to classic nearest-neighbor techniques for producing product recommendations [1] and has thus been attracting attention. Another approach is tensor factorization for tensor data consisting of users, items, and time [2].

If the data is sparse, however, similarities among users sometimes cannot be computed using CF because some items may not have been evaluated by multiple users. In such cases, an item cannot be recommended. An effective way to solve this problem is to concurrently use other information in addition to user ratings such as item metadata [3][4] and review text [5][6]. Review text is particularly useful for analyzing user preferences because it represents the contents of item evaluations and the impressions of users.

In addition, many recommendation services provide aggregated information such as the average of ratings by different users so that users can see the reputation of items at a glance. However, many recommendation services do not display aggregated information for items with few ratings because they are not reliable. Moreover, reviews are not always reliable because anyone can easily evaluate items, so some evaluation contents are malicious.

We investigated the effectiveness of six existing CF methods for large-scale sparse multicriteria rating data. We formulated rating aggregation as a CF problem and applied CF methods to it. Furthermore, we extended three of the existing methods to calculate user similarity using indirect user relationships and review comments and applied them to CF and rating aggregation. The results show that multicriteria rating approaches perform better than single criterion rating approaches, and the collaborative filtering methods using multicriteria rating predicted aggregated ratings more accurately than ones using single criterion rating. We compared the quality of our extended methods with the unextended methods and found that the extended methods performed better for predicting hotel ratings and aggregating hotel ratings.

The remainder of this paper is organized as follow. Section 2 provides an overview of the existing CF approaches that use similarity between users and of the matrix factorization and tensor factorization recommendation models. In Section 3, we introduce the rating aggregation problem and the several existing methods. In Section 4 presents our extended methods, which calculate user similarity using indirect users and review comments. In Section 5, we describe our experimental set up for evaluation and present and discuss the results. We conclude in Section 6 by summarizing the key points and mentioning future work.

II. COLLABORATIVE FILTERING

In this section, we first briefly explain the user-based CF methods, which predict an unknown evaluation rating for each item for each user from the ratings by similar users, using single criterion rating and introduce methods that use multicriteria rating. Then, we explain the matrix factorization and tensor factorization models.

A. Single Criterion Rating

The conventional approach to CF is based on the assumption of single-criterion rating and predicts the unknown rating of each item by calculating the similarity between users on the basis of their previous ratings. Cosine similarity (1) is used as the standard method to calculate the similarity between two different users. If $I(u,u')$ represents the set of all items rated by both users u and u',

$$\mathrm{sim}(u,u') = \frac{\sum_{i\in I(u,u')} R(u,i)R(u',i)}{\sqrt{\sum_{i\in I(u,u')} R(u,i)^2}\sqrt{\sum_{i\in I(u,u')} R(u',i)^2}}. \quad (1)$$

The predicted unknown rating for item i for user u, $\mathrm{pred}(u,i)$, is calculated using

$$\mathrm{pred}(u,i) = \overline{r_u} + \frac{\sum_{u'\in N} \mathrm{sim}(u,u')(r_{u'i} - \overline{r_u})}{\sum_{u\in N} |\mathrm{sim}(u,u')|}, \quad (2)$$

where N represents the set of users that are similar to user u and $\overline{r_u}$ represents the average rating of user u.

B. Multicriteria Rating

An extended method that deals with multicriteria ratings by using overall similarity or aggregation function has been proposed [7]. In this method, each rating by user u for item i consists of an overall rating r_0 and k multicriteria ratings:

$$\mathrm{R}(u,i) = (r_0, r_1, \cdots, r_k). \quad (3)$$

1) Overall Similarity: Assume a user evaluates $k+1$ different values. There are several methods for calculating the similarity using the multidimensional distance. If each rating the user gives to an item is a point in the $k+1$ dimensional space, the Chebyshev distance to can be used to calculate the multidimensional distance:

$$\max_{i\in\{0,\ldots,k\}} |r_i - r_i'|. \quad (4)$$

The overall similarity d_{user} is calculated using the overall distance:

$$\mathrm{sim}(u,u') = \frac{1}{1 + d_{\mathrm{user}}(u,u')}. \quad (5)$$

This definition of similarity has desirable range properties. The similarity will approach 0 as the distance between two users becomes larger, and it will be 1 if the distance is 0.

2) Aggregation Function: The aggregation approach to predict values is based on the assumption that the overall rating is not independent of the multicriteria ratings but rather is an aggregation of the ratings [7]. It comprises three steps.

Step 1: Predict unknown multicriteria rating for each individual criterion but not overall rating using any recommendation techniques.

Step 2: Learn aggregation function; i.e., learn relationship between overall rating and underlying multicriteria rating of each item.

Step 3: Predict overall rating of each item by using ratings for each individual criterion and aggregation function.

C. Matrix Factorization

One of the most commonly used approaches to CF is based on the matrix factorization model [1]. This approach characterizes both users and items by using vectors of latent factors inferred from a rating matrix. The matrix contains the evaluation rating for each item by each user. For example, for I users and J items, given $I \times J$ user-item rating matrix $R = [r_{ij}]_{I\times J}$, the matrix factorization model represents rating matrix R as the product of K-rank factors $R \approx U^t V$, where $U \in R^{K\times I}$ and $V \in R^{K\times J}$. Parameter K controls the number of latent factors for each user and item and is typically much smaller than I and J.

The latent representations of the users and items are computed by minimizing the following regularized squared error on the set of known ratings.

$$\min_{U,V} \sum_{(i,j)\in k} (r_{ij} - u_i^t v_j)^2 + \lambda(||u_i||^2 + ||v_j||^2), \quad (6)$$

where k is the set of user and item pairs for which r is the known rating, and constant λ controls the strength of regularization to avoid overfitting the observed evaluation rating.

D. Tensor Factorization

In matrix factorization, the relationship between two objects is modeled using a low-rank matrix. However, in some cases, the relationships among more than two objects are established. These relationships can be represented as a multidimensional array, which is a generalization of matrix factorization [2]. In our experiment, we used the relationships among users, items, and multicriteria. We used conditional probability (CP) factorization (CP:CANDECOMP/PARAFAC), a tensor factorization method, and decomposed a tensor into the sum of rank-one tensors:

$$\chi \simeq \sum_{k=1}^{K} u_k \circ v_k \circ w_k. \quad (7)$$

TABLE I
Rating set

	Item 1	Item 2	Item 3	Item 4
User a	5	2	?	?
User b	?	?	2	3
User c	3	4	4	3

Each element x_{ijk} of χ can be calculated using $x_{ijk} \simeq \sum_{k=1}^{K} u_{ik}v_{jk}w_{lk}$, where $U = (u_{ik}) = (u_k)_{k=1}^{K} \in R^{I\times K}$, $V = (v_{jk}) = (v_k)_{k=1}^{K} \in R^{J\times K}$, and $W = (w_{lk}) = (w_k)_{k=1}^{K} \in R^{L\times K}$.

III. Rating Aggregation

The simplest method for aggregating the ratings of an item by several users is to average the ratings. However, if the number of users is very small, the aggregated rating is affected by the ratings of specific users, and if the reliabilities of the ratings are low, the reliability of the aggregated rating is also low. For this reason, many Web sites do not display an aggregated rating if the number of evaluators is small.

Several studies on reliably aggregating ratings from people in general (“workers”in the parlance of crowdsourcing) have focused on binary or multi-class labeling. They include ones that considered worker ability [8], problem difficulty [9], and worker confidence [10]. Other studies have focused on multilabeling of data wherein a data item can have multiple labels at the same time [11]. Several studies have focused on the results of aggregating rating data [12]. The conventional approaches to rating aggregation are based on the premise that single rating, and these approaches use probabilistic model. In our experiment of the efficiency of CF of multicriteria data, we regarded rating aggregation as an information recommendation problem and considered the ”average user”.

IV. Extended Methods

To overcome the problem of calculating the similarity of two users when there is sparse rating data, we extended the existing CF methods so that they calculate user similarity using indirect users and review comments.

First, we explain how user similarity is calculated using indirect users. For example, consider users a and b who do not have a rated item in common, meaning that the similarity between them cannot be directly calculated. On the other hand, the similarities between users a and c and users b and c can be calculated because there are rated items in common between both pairs, as shown in (TABLE I). In this case, the similarity between users a and b is calculated on the basis of the similarity between users a and c and between users b and c:

$$\mathrm{sim}(a,b) = \frac{(\mathrm{sim}(a,c) + \mathrm{sim}(b,c))}{2}.$$

If there are two or more indirect users, the overall average from the indirect user set N is used:

$$\mathrm{sim}(a,b) = \frac{\sum_{u\in N} \mathrm{sim}(a,u) + \mathrm{sim}(b,u)}{2|N|}. \tag{8}$$

Next, we explain how user similarity is calculated using review comments. First, all review comments contributed by each user are gathered into one text string. Then, the parts of speech are extracted using morphological analysis, and the cosine similarity between the comments of different users is calculated from feature vectors weighted by the term frequency-inverse document frequency (TF-IDF). The similarity between all users can be calculated since review comments can be gathered from all contributing users regardless of whether there are any rated items in common.

$$\mathrm{tf}(t,d) = \frac{n_{ij}}{\sum_k n_{kj}} \tag{9}$$

$$\mathrm{idf}(t,d) = \log \frac{N}{1 + \mathrm{df}(t,d)} \tag{10}$$

$$\text{tf-idf}(t,d) = \mathrm{tf}(t,d)\times\mathrm{idf}(t,d), \tag{11}$$

where n_{ij} is the number of instances of word i in sentence j, $\sum_{\mathrm{k}} \mathrm{n_{kj}}$ is the total number of words in sentence j, N is the total of number of documents, and df is the number of documents containing word t. Because considering that there is a word that has never appeared, it will be 1 if df is 0 (10). A logarithm is used in (10) to prevent low-frequency documents being excessively weighted. When analyzing text data, it is normal for a word to appear frequently in several documents. Many of these frequently appearing word do not carry meaningful or identifying information. By using TF-IDF, we decrease the weight of these words and increase the weight of words appearing frequently in a specific sentence.

V. Experiment

A. Collaborative Filtering

1) Experimental Setup: To evaluate the performance of the six existing collaborative filtering methods and the extended versions, we used the Rakuten Travel dataset (https://rit.rakuten.co.jp/data_release/) for large-scale sparse multicriteria rating data. Each user provides review comments and an overall rating for a hotel along with ratings for six criteria: *location*, *room*, *meals*, *bath*, *service*, and *equipment* on a scale of 1 to 5. The dataset includes data from 801 users for 5,098 hotels, with 16,993 ratings in total. We randomly extracted four rating items for each user and used them as items with an unknown rating. We collected these items into a test data and predicted the overall ratings. Prediction accuracy was measured using the root mean squared error (RMSE).

The six existing methods included

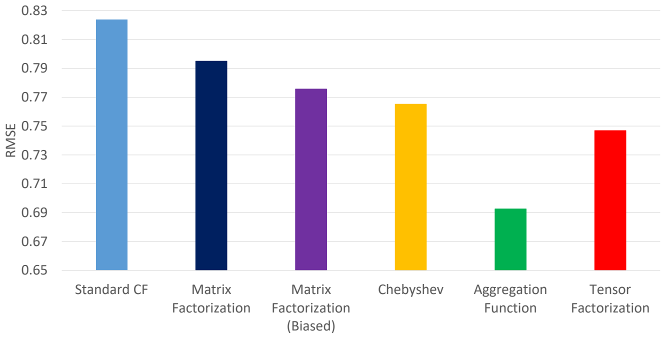

Fig. 1. RMSE for existing six unextended collaborative filtering methods

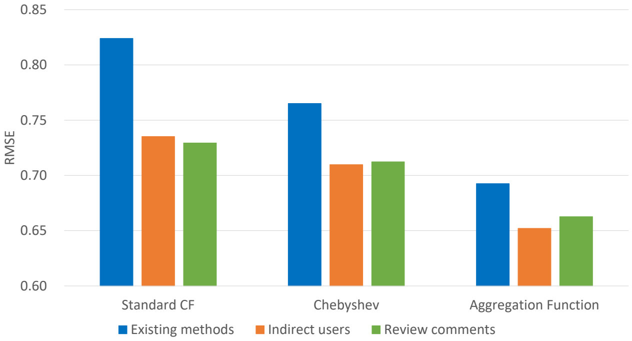

Fig. 2. RMSE for three existing methods that use similarity

Standard CF

This is the baseline method used to compare a single criterion rating approach to the multicriteria rating approaches. It calculates cosine similarity by using the overall rating.

Chebyshev

This method calculates similarity for multicriteria ratings by using the Chebyshev multidimensional distance metric.

Aggregation Function

This method uses an aggregation function based on linear regression. Standard CF is used to predict each individual criterion; the overall rating is predicted using the ratings for each criterion and the aggregation function.

Matrix Factorization

This method characterizes both items and users by vectors of factors that are inferred from the rating matrix. High correspondence between user and item factors leads to a recommendation.

Matrix Factorization (Biased)

This method adds biases to Matrix Factorization.

Tensor Factorization

This method is similar to Matrix Factorization except that it characterizes items, users, and multicriteria by using vectors of factors inferred from the rating matrix.

2) Results and Discussion: Fig. 1-2 summarizes the results for the six unextended methods.

First, as can be seen in Fig. 1, the methods that use similarity and multicriteria approaches (Chebyshev, Aggregation Function) had better performance than the one using the single criterion rating approach (Standard CF). Moreover, Matrix Factorization (Biased) had better performance than Matrix Factorization, and Tensor Factorization, which considers multicriteria, had better performance than the two matrix factorization methods. Therefore, it is more effective to consider multicriteria rating rather than single criterion rating when predicting ratings. Standard CF had a RMSE of 0.823, while Matrix Factorization had a RMSE of 0.795 and Matrix Factorization (Biased) had a RMSE of 0.775. The matrix factorization methods were thus the most effective of the similarity methods. The Chebyshev method had a RMSE of 0.765 while Tensor Factorization had a RMSE of 0.746. This means that Tensor Factorization was more effective than the similarity methods.

As can be seen in Fig. 2, the extended versions of the similarity methods had much better performance than the unextended methods. In general, although similar users are important in CF, the similarity between two users cannot be calculated if the rating data are sparse. To increase the number of similar users, we extended the existing methods by calculating user similarity using indirect users and review comments. As a result, the methods that use cosine similarity saw an increase in the number of people from about 1.5 to 8.2 while the methods that use Chebyshev similarity saw an increase from about 3.1 to 10.2. As a result, the extended methods performed much better than the unextended methods.

Finally, Aggregation Function performed the best of all methods. The reason for this is that, although the amount of information from each user for each item that can be used is small for large-scale sparse multicriteria rating, Aggregation Function learns the information for all users and items.

B. Rating Aggregation

1) Experimental Setup: To evaluate the performance of rating aggregation, we applied various CF methods to rating aggregation. The Rakuten Travel dataset does not have aggregated ratings. Therefore, we extracted the data for users who had evaluated 15 or more hotels and for hotels that had been evaluated by 15 or more users from the Rakuten Travel dataset to make aggregated ratings. The aggregated rating for each item was the average of the overall ratings of the users who had evaluated the item among the extracted users and hotels. We assumed that these aggregated ratings could be highly trusted and be used as a gold standard because these items were evaluated by many users. Then, we added the *average user* giving the aggregated ratings to the dataset. We regarded rating aggregation as an item recommendation problem to the *average user*, and evaluated prediction accuracy by using the CF between randomly selected known users and the *average user* to calculate the aggregated ratings from the evaluation of a small number of users. To measure prediction accuracy, we again used the RMSE.

2) Results and Discussion: Fig. 3-4 show the performance of the different methods. We used the same methods as for the CF experiment. First, as can be seen in Fig. 3, the multicriteria approaches (Chebyshev, Aggregation Function) had better performance than the single criterion rating approach (Standard CF), as in the CF experiment. We also compared Tensor Factorization to Matrix Factorization and Matrix Factorization (Biased). Matrix Factorization (Biased) had better performance than Matrix Factorization. Tensor Factorization, which considers multicriteria, had better performance than the matrix factorization methods. From this, and also in rating aggregation, it is effective to consider multicriteria rating rather than single criterion rating. Tensor Factorization also had better (or similar) performance than the Standard CF.

Finally, we compare the performance of the extended and unextended methods, as presented in Fig. 4. We formulated rating aggregation as a CF problem for the average user. Because the average user evaluates all items, there is no problem in calculating user similarities. Therefore, we used only review comments in the extended methods. The extended versions of the similarity methods perform much better than the unextended methods when the number of known users was small.

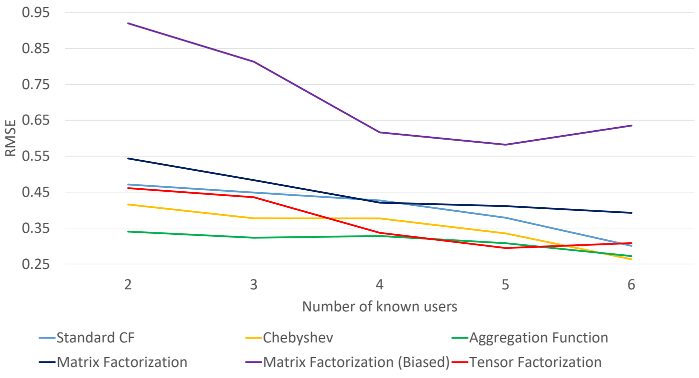

Fig. 3. RMSE for existing six unextended collaborative filtering methods

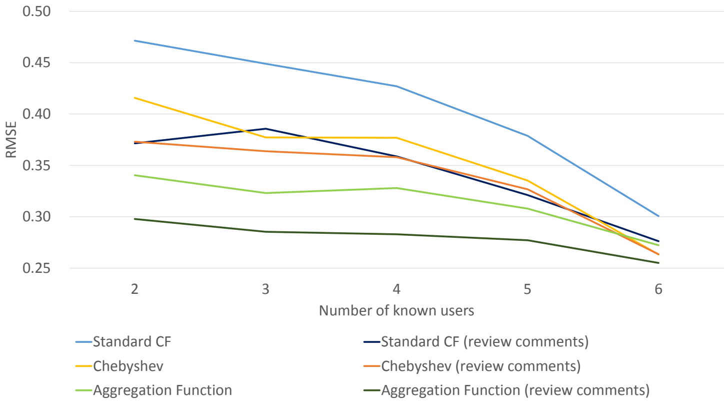

Fig. 4. RMSE of extended and unextended methods

VI. Conclusion

In this paper, we investigated the effectiveness of existing collaborative filtering methods for large-scale sparse multicriteria rating data. We found that multicriteria rating approaches perform better than single criterion rating approaches. We formulated rating aggregation as a collaborative filtering problem and applied six collaborative filtering methods to it. We found that the collaborative filtering methods using multicriteria rating predicted aggregated ratings more accurately than ones using single criterion rating. We extended three of the existing methods to calculate user similarity using indirect users and review comments and applied them to collaborative filtering and rating aggregation. The extended methods had better performance both in collaborative filtering and in rating aggregation.

Future work includes using Latent Dirichlet Allocation (LDA) [13] to analyze the review comments rather than the bag-of-words (BoW) model. We will also investigate using other types of models such as probabilistic matrix factorization and probabilistic tensor factorization [2][14].

Acknowledgments

We thank the anonymous reviewers for their helpful comments. We used Rakuten Travel data provided for research purposes by the National Institute of Informatics in cooperation with Rakuten, Inc. This work was partially supported by JSPS KAKENHI Grant Number JP15H02782.

References

[1] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, 2009.

[2] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, “Temporal collaborative filtering with bayesian probabilistic tensor factorization,” in *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 2010, pp. 211–222.

[3] D. Agarwal and B.-C. Chen, “flda: matrix factorization through latent dirichlet allocation,” in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 91–100.

[4] S. Feng, J. Cao, J. Wang, and S. Qian, "Recommendations based on comprehensively exploiting the latent factors hidden in items'ratings and content," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 3, p. 35, 2017.

[5] Y. Bao, H. Fang, and J. Zhang, "Topicmf: Simultaneously exploiting ratings and reviews for recommendation." in *AAAI*, vol. 14, 2014, pp. 2–8.

[6] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 165–172.

[7] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48–55, 2007.

[8] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.

[9] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS*, 2009, pp. 2035–2043.

[10] S. Oyama, Y. Baba, Y. Sakurai, and H. Kashima, "Accurate integration of crowdsourced labels using workers' self-reported confidence scores." in *IJCAI*, 2013, pp. 2554–2560.

[11] L. Duan, S. Oyama, H. Sato, and M. Kurihara, "Separate or joint? estimation of multiple labels from crowdsourced annotations," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5723–5732, 2014.

[12] J. S. Uebersax and W. M. Grove, "A latent trait finite mixture model for the analysis of rating agreement," *Biometrics*, pp. 823–835, 1993.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[14] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.