Situation-Aware Multi-Criteria Recommender System: Using Criteria Preferences as Contexts

Yong Zheng
School of Applied Technology
Illinois Institute of Technology
Chicago, Illinois, USA
yzheng66@iit.edu

ABSTRACT

Recommender systems (RSs) have been successfully applied to alleviate the problem of information overload and assist users' decision makings. Multi-criteria recommender systems is one of the RSs which utilizes users' multiple ratings on different aspects of the items (i.e., multi-criteria ratings) to predict user preferences. Traditional approaches simply treat these multi-criteria ratings as addons, and aggregate them together to serve for item recommendations. In this paper, we propose the novel approaches which treat criteria preferences as contextual situations. More specifically, we believe that part of multiple criteria preferences can be viewed as contexts, while others can be treated in the traditional way in multi-criteria recommender systems. We compare the recommendation performance among three settings: using all the criteria ratings in the traditional way, treating all the criteria preferences as contexts, and utilizing selected criteria ratings as contexts. Our experiments based on two real-world rating data sets reveal that treating criteria preferences as contexts can improve the performance of item recommendations, but they should be carefully selected. The hybrid model of using selected criteria preferences as contexts and the remaining ones in the traditional way is finally demonstrated as the overall winner in our experiments.

CCS Concepts

•**Information systems** → **Personalization; Recommender systems;** *Collaborative filtering;*

Keywords

recommender system; context; multi-criteria; decision making

1. INTRODUCTION AND RELATED WORK

Recommender systems (RSs) are able to assist users' decision making by providing personalized item recommendations tailored to their preferences. During the last decade, several novel types of RS have been developed. One of these novel RSs is multi-criteria recommender systems (MCRS) [9, 11, 2], where users leave not only an *overall rating* on items, but also multiple ratings on different attributes or aspects (i.e. *multi-criteria ratings*) of the items. As the tourism example shown in Figure 1, users can additionally leave ratings on 7 different aspects of the hotels: value for the money, quality of rooms, convenience of the hotel location, cleanliness of the hotel, experience of check-in, overall quality of service and particular business services (such as business center, meeting rooms, computers, printers, etc). We can also find related applications in other areas, such as in the movie domain, users can rate the story of the movie and visual effects in addition to an overall rating on the movie itself. In restaurant area, users can additionally rate the food, ambience, service, and so forth. Accordingly, the utility function of how users like items in MCRS is no longer a function with a single rating (i.e., the overall rating), but also the multi-criteria ratings which can represent more complex user preferences.

Figure 1: Example of Multi-criteria ratings in TripAdvisor

More specifically, traditional recommender systems only take users' ratings on the items into consideration, as shown by Equation 1. R_0 is the overall rating given by a user on one item.

$$R : Users \times Items \rightarrow R_0 \quad (1)$$

MCRS additionally consider user's ratings on different aspects of the items, as shown by Equation 2, where we assume there are k aspects of the items (such as room size, hotel cleanliness, experience of check-in in the hotel recommendation), and users may give ratings to each aspect in addition to the overall rating R_0.

$$R : Users \times Items \rightarrow R_0 \times R_1 \times R_2 \times ... \times R_k \quad (2)$$

An example of the multi-criteria rating data can be shown by Table 1. The *rating* in the table is equivalent to users' overall rating on the items. We also have users' ratings on multiple criteria, such as room, check-in and service.

SAC 2017,April 03-07, 2017, Marrakech, Morocco

http://dx.doi.org/10.1145/3019612.3019918

Table 1: Example of Rating Data from TripAdvisor

User	Item	Rating	Room	Check-in	Service
U_1	T_1	3	3	4	3
U_2	T_2	4	4	4	5
U_3	T_1	?	?	?	?

The standard problem in multi-criteria recommendation, therefore, can be summarized as how to take multi-criteria ratings into consideration to better predict how a user may like an item. In other words, given a user U_3 and an item T_1 shown in the table above, how to predict U_3's overall rating on T_1. Note that we also do not know U_3's multi-criteria ratings on T_1.

Several research has made their contributions to multi-criteria recommendations by improving the popular collaborative filtering (CF) recommendation technique. One of the popular methods is the heuristic approach [1, 10] which utilize the multi-criteria ratings to better calculate user-user or item-item similarities in the neighborhood-based CF algorithms. Another one is the model-based approach [1, 12, 8] which constructs a predictive model to estimate a user' overall rating on one item from the observed multi-criteria ratings.

Context-aware recommender systems (CARS) [3] is another type of RS which take contexts into consideration and try to adapt recommendations to users' contextual situations. The notion of the context can be viewed as the dynamic variables which may change during activities or user interactions [14], e.g, time, location, companion, etc. The underlying assumption in CARS is that user's preferences on items may vary from contexts to contexts. For example, a user may choose a difference movie if he or she is going to watch the movie with *partner* rather than with *kids*.

Traditionally, the approaches for multi-criteria recommender systems only treat the item aspects as multiple criteria and aggregate them together to predict user preferences. Alternatively, the preferences on these criteria could be able to be viewed as context information. For example, assuming a user is viewing the hotel reviews on TripAdvisor.com. He or she may have judgements on different aspects of the hotels by reference to other users' perspectives. In terms of the psychology, the user may already "rate" the multiple hotel criteria, such as room - 5 star (i.e., very good), business - 3 star (i.e., acceptable). And user will make a final decision about whether they will reserve this hotel. In this case, the users' judgements on different aspects of the hotel (i.e., criteria preferences) can be viewed as the context or the situation in which user will make a final decision. We prefer to give the name "*Situation-aware Multi-criteria Recommender Systems*" to this type of RS, where we can treat the users' preferences on multiple criteria as contexts. In this paper, we explore the impacts of treating criteria preferences as contexts in comparison with using criteria in a traditional way (i.e., aggregation-based multi-criteria item recommendations).

2. PROBLEM STATEMENT

As mentioned before, there are different ways to build multi-criteria recommendation algorithms. The aggregation-based approach [1, 6] is popular for its ease and effectiveness. In this section, we introduce this approach first and then point out our problem statements.

2.1 Preliminary: Aggregation-Based Approach

The aggregation-based approach builds an aggregation function [1, 6] f (shown in Equation 3) that represents the relationship between the overall rating R_0 and multi-criteria ratings (e.g., R_1, R_2, ..., R_k), in order to aggregate the multi-criteria ratings to estimate how a user will like an item.

$$R_0 = f(R_1, R_2, ..., R_k) \tag{3}$$

There could be different ways to generate the function f. We use the linear aggregation [1] described by Equation 4.

$$R_0 = w_1 * R_1 + w_2 * R_2 + ... + w_k * R_k + t \tag{4}$$

This approach assumes there is a linear relationship between the multi-criteria ratings and the overall rating. R_0 can be estimated by a multiple linear regression, where we assign a weight (e.g., w_k) to each criterion, and finally learn these weights, as well as the intercept (i.e., t in Equation 4) by minimizing the squared prediction errors. More specifically, we choose support vector regression (SVR) [13, 5] which is proved to work better than lasso and ridge regressions in these aggregation-based approaches.

Usually, such a simple linear regression model may not work well, because it produces a same linear function without considering users and items. By taking personalization into account, we can create these linear aggregation functions for each user, which is a *user-specific* linear aggregation model. Or, we can also create these linear aggregation functions for each item, which is a *item-specific* linear aggregation model. Jannach, et al. [5] proposed to combine user-specific and item-specific models together by a linear weighted approach. In this paper, we follow the hybrid approach which combines the user-specific and item-specific linear aggregation models, as the baseline in our work.

Therefore, we choose the linearly weighted combination of user-specific and item-specific linear aggregation model as the base algorithm for multi-criteria recommendation. Then the multi-criteria recommendation algorithm (either user-specific or item-specific) can be described in three steps:

1. **Aggregations:** We learn the parameters in the linear regression functions (shown in Equation 4) during the training.
2. **Multi-criteria Rating Predictions:** To predict the overall rating given by a user on one item in the test data set, we first estimate the k individual ratings (such as the ratings on room, check-in and service shown in Table 1) using any traditional recommendation algorithms. Namely, the k-dimensional multi-criteria rating problem in this step is decomposed into k single-rating recommendation problems. Ratings in each criterion along with user and item information can be used as an individual rating data for rating prediction purpose (i.e., predict ratings for each criterion). In our experiment, we use biased matrix factorization (BiasedMF) [7] as the recommendation algorithm to estimate a user's rating on each criterion.
3. **Overall Rating Predictions:** Finally, we incorporate the k predicted ratings into the aggregation function learned in step 1 to estimate how a user may like an item (the overall rating).

2.2 Research Problems

Typically, there are two ways to improve the aggregation-based multi-criteria recommendation algorithms introduced above. On one hand, it is important to improve the rating predictions on each criterion, since the final user preference is estimated based on these multiple rating predictions. On the other hand, it is useful to explore different ways to utilize these predicted multi-criteria ratings.

In this paper, we focus on the second work – exploring different ways to utilize the predicted multi-criteria ratings.

We believe user preferences on multiple criteria can be viewed as contextual situations. For example, assuming a user is viewing the hotel reviews on TripAdvisor.com. He or she may have judgements on different aspects of the hotels by reference to other users' perspectives. In terms of the psychology, the user may already "rate" the multiple hotel criteria, such as room - 5 star (i.e., very good), business - 3 star (i.e., acceptable). And user will make a final decision about whether they will reserve this hotel. In this case, the users' judgements on different aspects of the hotel (i.e., criteria preferences) can be viewed as the context or the situation in which user will make a final decision. In other words, the overall rating predictions can be formulated to such a problem: given the contextual situation – how much a user likes different aspects of the items, the system will predict whether the user will like this item or not.

More specifically, we'd like to compare the recommendation performance among three settings: using all the criteria ratings in the traditional way, treating all the criteria preferences as contexts, and utilizing selected criteria ratings as contexts.

3. METHODOLOGIES

Therefore, we are going to explore whether the recommendation performance can be improved if we use criteria preferences as contexts. We are able to produce three models: full, partial and hybrid contextual models. We introduce them respectively as follows.

3.1 Full Contextual Model (FCM)

We utilize the criteria preferences as contexts by these steps:

1. **Multi-criteria Rating Predictions:** As mentioned before, our goal is to seek improvements by different ways to utilize multi-criteria ratings, instead of better predicting these ratings. This step is the same operation as the second step in the original aggregation-based approach. We use BiasedMF algorithm to predict users' ratings on each criteria.
2. **Overall Rating Prediction by Full Contextual Model:** In FCM, all of the criteria preferences are viewed as contexts, while they are indicated by the predicted multi-criteria ratings in the first step. Recall that the predicted ratings are double values with decimals. However, each rating value will be treated as categories in CARS. For example, rating 5 means "very good", 4 indicates "good", 3 is "acceptable", 2 tells "bad" and 1 indicates "very bad". Rating values with decimals will introduce context sparsity issues. In this case, we use either casting or rounding to convert the predicted multi-criteria ratings to integers. In our experiments, rounding is the better solution. Afterwards, we can apply any context-aware recommendation algorithms to predict a user's overall rating on the items. For evaluation purpose, we choose context-aware matrix factorization (CAMF) [4] in this work. More specifically, we choose the CAMF_C approach which will learn the rating deviations in each context that are embedded into the matrix factorization algorithms.

3.2 Partial Contextual Model (PCM)

Apparently, not all of the criteria preferences should be considered as context. For example, *service* and *value* may be influential, but *business* may be not that important. Using irrelevant variables as contexts in the CAMF model may introduce noises and finally result in bad recommendation performance. Alternatively, we can select the most influential criteria as contexts. In our work, we use information gain based on the entropy as the impurity metric to select the top influential criteria as contexts. We continue to use these selected criteria as contexts in CAMF_C and ignore the preferences in other criteria to predict users' overall rating on the items.

3.3 Hybrid Contextual Model (HCM)

The HCM is an improvement over the PCM. We use selected criteria as contexts to be applied in the CAMF_C approach to build the PCM. And we use the remaining criteria to build the aggregation-based multi-criteria recommendation model described in Section 2.1. Finally, we develop a linear hybrid model to combine these two models and produce the final overall rating predictions.

4. EXPERIMENTAL EVALUATIONS

4.1 Evaluation Settings

There are not many data sets with multi-criteria ratings for public research. In our paper, we use these two real-world data sets:

- **TripAdvisor data:** This data was collected by Jannach, et al. [5] through a Web crawling process which records users' hotel ratings on hotels located in 14 global metropolitan destinations, such as London, New York, Singapore, etc. There are 22,130 ratings given by 1,502 users and 14,300 hotels. Each user gave at least 10 ratings which are associated with multi-criteria ratings on seven criteria: value for the money, quality of rooms, convenience of the hotel location, cleanliness of the hotel, experience of check-in, overall quality of service and particular business services.
- **YahooMovie data:** This data was obtained from Yahoo!Movies website by Jannach, et al. [5]. There are 49,351 ratings given by 2,162 users on 3,075 movies. Each user left at least 10 ratings which are associated with multi-criteria ratings on four criteria: direction, story, acting and visual effects.

For each data, we use 80% of the ratings as training, and 20% as testing set. We evaluate and compare the algorithms mentioned above based on the rating prediction task. We predict the overall ratings given by each user on the item in the test set and measure the performance by mean absolute error (MAE). We use the CAMF_C implemented in CARSKit [15] to proceed with experiments.

4.2 Experimental Results

The experimental results in our work are presented in Figure 2, where the data labels on the top of each bar present the percentage of improvement by HCM in comparison with other algorithms.

Among these algorithms, BiasedMF describes the results which are produced by the BiasedMF algorithm using the overall ratings only. Agg presents the results based on the aggregation-based approach (described in Section 2.1) which takes advantage of multi-criteria ratings. Note Agg is the hybrid model that combines user-specific and item-specific aggregation models. FCM, PCM and HCM are the proposed models in this paper. In PCM, we select the most influential criteria as contexts using information gain. We tried different selections and combination of our selections. We only present the best selections due to limitations on the length of the paper. *Value* and *Room* are selected in the TripAdvisor data, while *Direction* and *Story* are selected in the YahooMovie data. HCM is the hybrid model of PCM and aggregation-based approach which utilize the criteria that are not selected as contexts.

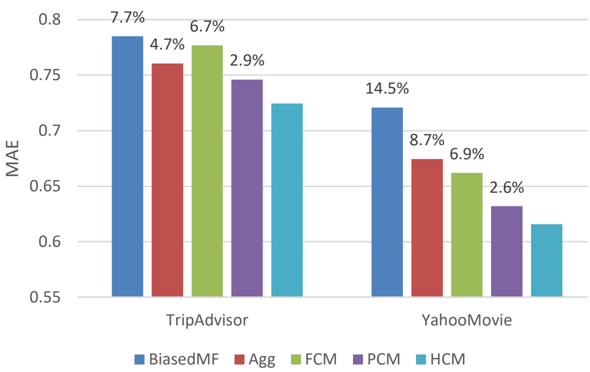

Figure 2: Experimental Results Based on MAE

First of all, BiasedMF is the worst model since it does not take any extra information (such as multi-criteria ratings or contexts) into consideration. Using all criteria preferences as contexts may not always be a good choice, since FCM performs worse than the Agg approach in the TripAdvisor data. By selecting the most influential criteria, PCM is able to outperform Agg in these two data sets. Finally, we can observe that HCM is the best predictive model which obtains the lowest MAE. It is able to offer significant (based on the statistical paired t-test) improvements compared with other models. More specifically, it is able to obtain 4.7% and 8.7% improvements in comparison with the Agg model, 6.7% and 6.9% improvements compared with the FCM, in the TripAdvisor and YahooMovie data sets respectively. By a linear hybridization of PCM and the aggregation-based recommendation models, HCM is able to outperform PCM in our experiments.

5. CONCLUSIONS AND FUTURE WORK

We believe that users' preferences on multiple criteria can be utilized as contexts to formulate situation-aware multi-criteria recommender systems. In this paper, we explore different ways to treating criteria preferences as contexts in order to seek improvements over the aggregation-based multi-criteria recommendation models. We compare the recommendation performance among three settings: using all the criteria ratings in the traditional way, treating all the criteria preferences as contexts, and utilizing selected criteria ratings as contexts. Our experiments based on the TripAdvisor and YahooMovie rating data sets reveal that treating criteria preferences as contexts can improve the performance of item recommendations, but they should be carefully selected. The hybridization of partial contextual model and the aggregation-based recommendation models is demonstrated as the overall winer in our experiments. This paper focuses on the usage of multi-criteria ratings by ignoring the process of improving the quality of predicted multi-criteria ratings. Existing successful practice, such as the similarity of contexts [16], could be used to further obtain improvements. We will seek the solutions to this problem in our future work.

6. REFERENCES

[1] G. Adomavicius and Y. Kwon. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22(3):48–55, 2007.

[2] G. Adomavicius, N. Manouselis, and Y. Kwon. Multi-criteria recommender systems. In *Recommender systems handbook*, pages 769–803. Springer, 2011.

[3] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3):67–80, 2011.

[4] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 301–304. ACM, 2011.

[5] D. Jannach, M. Zanker, and M. Fuchs. Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations. *Information Technology & Tourism*, 14(2):119–149, 2014.

[6] T. Jhalani, V. Kant, and P. Dwivedi. A linear regression approach to multi-criteria recommender system. In *International Conference on Data Mining and Big Data*, pages 235–243. Springer, 2016.

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

[8] Q. Li, C. Wang, and G. Geng. Improving personalized services in mobile commerce by a novel multicriteria rating approach. In *Proceedings of the 17th ACM conference on World Wide Web*, pages 1235–1236, 2008.

[9] N. Manouselis and C. Costopoulou. Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10(4):415–441, 2007.

[10] N. Manouselis and C. Costopoulou. Experimental analysis of design choices in multiattribute utility collaborative filtering. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(02):311–331, 2007.

[11] P. Poompuang and W. Premchaiswadi. User and item pattern matching in multi-criteria recommender systems. In *Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010 11th ACIS International Conference on*, pages 20–25. IEEE, 2010.

[12] N. Sahoo, R. Krishnan, G. Duncan, and J. Callan. Research note-the halo effect in multicomponent ratings & its implications for recommender systems: The case of yahoo! movies. *Information Systems Research*, 23(1):231–246, 2012.

[13] A. Smola and V. Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.

[14] Y. Zheng. A revisit to the identification of contexts in recommender systems. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, pages 133–136. ACM, 2015.

[15] Y. Zheng, B. Mobasher, and R. Burke. CARSKit: A Java-based context-aware recommendation engine. In *2015 IEEE International Conference on Data Mining Workshop*, pages 1668–1671, 2015.

[16] Y. Zheng, B. Mobasher, and R. Burke. Integrating context similarity with sparse linear recommendation model. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 370–376. Springer, 2015.