Criteria Chains: A Novel Multi-Criteria Recommendation Approach

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 usually predict ratings on each criterion individually and aggregate them together to estimate the user preferences. In this paper, we propose an approach named as "Criteria Chains", where each combination of the criteria can be utilized in a way of contextual situations in order to better predict the multi-criteria ratings. Our experimental results based on the TripAdvisor and YahooMovies rating data sets demonstrate that our proposed approach is able to improve the performance of multi-criteria item recommendations.

ACM Classification Keywords
H.3.3 Information Search and Retrieval: Information filtering

Author Keywords
multi-criteria, context-awareness, recommender systems

INTRODUCTION AND MOTIVATIONS
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 RSs have been developed. One of them is multi-criteria recommender systems (MCRS) [8, 10, 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. Take hotel ratings on TripAdvisor.com shown in Figure 1 for example, 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 business services. It's also popular in other areas: users can rate the story of the movie and visual effects in addition to an overall rating on the movie itself; users can additionally rate the food, ambience, service in the restaurant domain. Accordingly, the utility function of how a user likes an item in MCRS is no longer a function only with the overall rating, but also the multi-criteria ratings which can represent complex user tastes.

IUI 2017, March 13-16, 2017, Limassol, Cyprus

DOI: http://dx.doi.org/10.1145/3025171.3025215

Figure 1. Example of Multi-criteria ratings in TripAdvisor

More specifically, traditional RSs 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.

$$R : Users \times Items \rightarrow R_0 \tag{1}$$

MCRS additionally consider user's ratings on different aspects of the items, as shown by Equation 2. We assume there are k aspects of the items, and users may give ratings to each aspect in addition to the overall rating R_0.

$$R : Users \times Items \rightarrow R_0 \times R_1 \times R_2 \times \ldots \times R_k \tag{2}$$

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.

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 problem in multi-criteria recommendation, therefore, can be summarized as how to additionally take multi-criteria ratings into account 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, 9] 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, 7] which constructs a predictive model to estimate a user' overall rating on one item from the observed multi-criteria ratings.

The approaches in multi-criteria recommender systems usually predict a user's ratings on each criterion respectively and try to make use of these predicted ratings for recommendation purposes. However, traditional approaches treat different criterion individually and ignore their dependencies, where Sahoo et al. [12] argued that the dependency among multiple criteria is able improve recommendations. In this paper, we propose a novel approach, named as "Criteria Chains", which takes the dependency among multiple criteria into consideration. More specifically, we construct the list of criteria as a chain and try to predict a user's rating on each criterion one by one by using the previous predictions as context information.

PRELIMINARY: AGGREGATION-BASED APPROACH

As mentioned before, there are different ways to build multi-criteria recommendation algorithms. The aggregation-based approach [1, 5] is popular for its ease and effectiveness. In this paper, we choose this approach as the base method and try to apply Criteria Chains on top of it. The idea of Criteria Chains could be general and applied to other multi-criteria recommendation algorithms.

The aggregation-based approach builds an aggregation function [1, 5] 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, 4] 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. [4] 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 linear weighted combination of user-specific and item-specific linear aggregation model as the base algorithm for multi-criteria recommendation. More specifically, the multi-criteria recommendation algorithm that built upon the aggregation-based approach (either user-specific or item-specific) can be described in three steps:

1. **Aggregations:** We learn the parameters in the linear multiple regression functions (shown in Equation 4) based on the training data set.
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 individual and independent 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. For example, we will use user-item-rating on service as the training set to predict a user's rating on the criterion "service" by given a hotel in the testing set. In our experiment, we use biased matrix factorization (BiasedMF) [6] as the recommendation algorithm to estimate a user's rating on each criterion.
3. **Overall Rating Predictions:** Finally, we can incorporate the k multi-criteria predicted ratings into the aggregation function learned in step 1 to estimate how a user may like an item (i.e., the overall rating).

RESEARCH PROBLEMS AND METHODOLOGIES

As mentioned previously, our proposed approach, *Criteria Chains*, will try to incorporate dependencies among multi-criteria ratings into the prediction process. We learn the experience from the *classifier chains* [11] which is one of the approaches used in the multi-label classification (MLC) task. The idea behind classifier chains is that it is able to learn the binary predictions (0 or 1) for each label in shape of a chain in the classification task. At the beginning, it uses the features in the data to make predictions for the first label. And this prediction will be added as an extra feature to the data in order to predict the second label. They gradually add the label predictions as additional features to estimate the next label, until all the labels have been predicted. The classifier chain is able to outperform several traditional MLC algorithms since it takes label dependencies into consideration.

Criteria chains can be used in a similar way. Take the TripAdvisor data shown in Table 1 for example, there are three criteria: Room, Check-in and Service. First of all, we will define a sequence for the chain, e.g., "Room - Check-in - Service". Then we predict U_3's rating on Room for item T_1 first, based on the rating matrix built for user, item, ratings on the criterion Room. Next, we take the predicted U_3's rating on Room for item T_1 as inputs to predict his or her rating on Check-in for item T_2, since Check-in lies in the second position in the pre-defined criteria chain. Finally, we take the predicted ratings on Room and Check-in as inputs to predict U_3's rating on Service for item T_1.

One of the challenges in criteria chains become how to take predicted criteria ratings as inputs in the recommendation process. The short answer to the first question is that we view criteria preferences as contextual information. For example, assuming a user is reading the hotel reviews on TripAdvisor.com, he or she may already "rate" the hotel aspects by reference to other users' perspectives. And the user will make a final decision about whether he or she 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 contextual situations in which user will make a final decision.

Accordingly, in criteria chains, the rating prediction for the next criterion can be transformed to such a problem: given the contextual situation which is represented by a set of predicted criteria ratings, how much the user likes the next aspect of the items. For example, given the situation that a user likes the Room and Check-in very much (i.e., 5 star on these two criteria), how the user will rate the Service for this hotel. By viewing criteria preferences as contexts, we can use any context-aware recommendation algorithms to take these predicted criteria ratings into account and make predictions for a user's rating on the remaining criteria respectively. In our paper, we use context-aware matrix factorization (CAMF) [3][1] as the context-aware recommendation algorithms. In context-aware recommendation models, each rating value will be treated as categories. For example, rating 5 means "very good", 4 indicates "good", 3 is "acceptable", 2 tells "bad" and 1 indicates "very bad". The predicted ratings for each criterion are actually double values with decimals, which will introduce context sparsity issues. In this case, we use either casting or rounding to convert the predicted multi-criteria ratings to integers. And in our experiment, we found casting was the best solution.

By using criteria preferences as context information, we are able to build three models:

- **Criteria Chains: Aggregation Model (CCA):** We adopt criteria chains to predict the multiple criteria ratings one by one based on the CAMF_C algorithm. And finally we continue to use the aggregation-based approach to build a linear hybrid of use-specific and item-specific aggregation models based on SVR.

- **Criteria Chains: Contextual Model (CCC):** Alternatively, we continue to consider the predicted multiple criteria ratings (processed by casting) as contextual situations, and predict users' overall ratings on the items based on the CAMF_C approach.

- **Criteria-Independent Contextual Model (CIC):** We predict criteria ratings individually without considering dependencies among multiple criteria by using the BiasedMF algorithm, and finally view all of these predictions as contexts to predict the user's overall rating on the items based on the CAMF_C approach.

Note that both CCA and CCC take dependencies among multiple criteria into consideration, while the only difference is how to finally utilize the rating predictions to estimate the overall rating on the items. CIC [2] is considered as the baseline approach which ignores the criteria dependencies.

Another challenge is how to well-define the sequence of criteria in the chains. The sequence matters because the next prediction may be not that accurate if the previous predictions result in higher errors. We propose three different strategies to produce the sequence of criteria in criteria chains: 1). *random approach* which generates the sequence randomly; 2). *by errors*. We predict multiple criteria ratings individually by assuming they are independent, and then rank the criteria by the errors in ascending order; 3). *by information gain*. We produce the sequence by calculating the information gain based on entropies, while we use multiple criteria ratings as features and treat the value of the overall ratings as labels in the training set. For every strategy, we predict the rating on the first criterion based on the BiasedMF approach over the user-item-ratings on this criterion data set. And the rating predictions on other criteria will follow the criteria chains.

EXPERIMENTAL EVALUATIONS

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

- **TripAdvisor data:** This data was crawled by Jannach, et al. [4]. The data was collected through a Web crawling process which collects users' 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 YahooMovies by Jannach, et al. [4]. 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.

We use a 5-folds cross validation for each data and evaluate the recommendation performance by mean absolute error (MAE), precision and recall. Due to the limited space in this paper, we only present the results based on MAE. The results based on precision and recall can reveal similar patterns.

[1]We use CAMF_C which is the best option in our experiments.

[2]CIC is equivalent to the FCM approach described in [15].

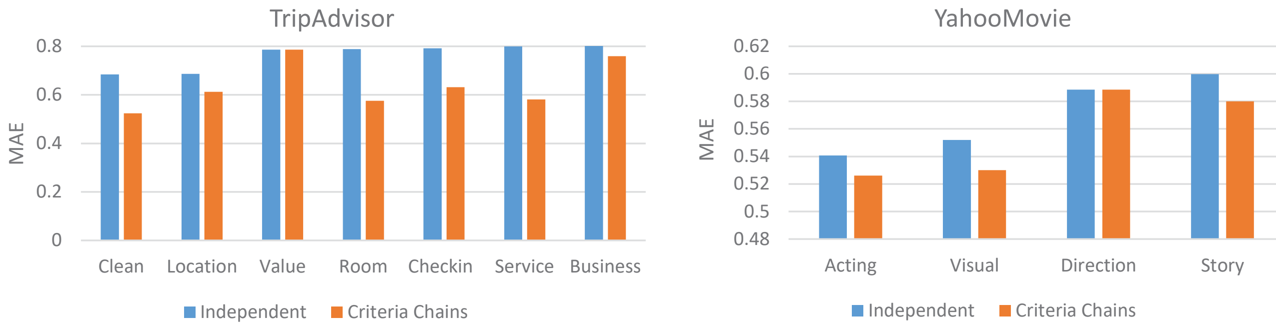

Figure 2. MAE by Multiple Criteria

The experimental results are described by Figure 3, where the data labels on the top of each bar present the percentage of improvement by CCC 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) which takes advantage of multi-criteria ratings. CCA and CCC are the two approaches that utilize criteria chains, while CIC is the approach that ignores criteria dependencies.

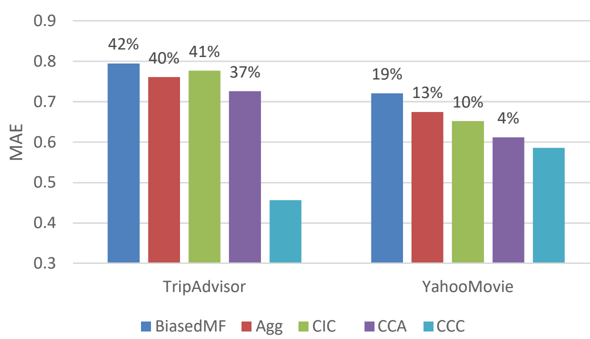

Figure 3. Experimental Results Based on MAE

Apparently, CCA and CCC can outperform other approaches based on the results. More specifically, criteria chains is adopted in CCA and CCC, and CCC is the best approach, which tells that viewing criteria tastes as contexts is better than aggregating them by SVR. Both CIC and CCC utilize CAMF_C to predict the overall rating by viewing criteria preferences as contexts, while CIC ignores the criteria dependencies. Based on the results, we can see that the quality of predicted multi-criteria ratings by criteria chains is much better, which further results in lower MAE by CCC.

In order to further understand the advantages of criteria chains, we present the MAE by each criterion in Figure 2. The blue bar presents the MAE based on the predictions in the traditional way – we assume the multiple criteria are independent. The red bar indicates the MAE based on criteria chains, where we use the information gain to produce the sequence of criteria and try to make predictions of ratings in each criterion by criteria chains. We can see that using criteria chain is able to help obtain the lower MAE, except the "Value" in the TripAdvisor and "Direction" in the YahooMovie data. It is because they are the first criterion in the chains for these two data respectively. There are much more improvements based on the MAE values by each criterion in the TripAdvisor data than the ones in the YahooMovie data – that's the reason why CCC is finally able to obtain much more improvement in the Tripadvisor data (around 40%) than in the YahooMovie data (around 13%) compared with the baseline approaches.

Furthermore, we explore which ordering strategy is the best option to produce the sequence of criteria in the chains. From Table 2, we can observe that using information gain (based on entropies) is the best solution in our experiments. It is not that surprising because entropy is usually used as the impurity metric to determine which context variable is more influential.

Table 2. MAE Comparison Among Different Ordering Strategies

	TripAdvisor	YahooMovie
Random	0.81	0.77
by Error	0.78	0.72
by IG	0.46	0.59

CONCLUSIONS AND FUTURE WORK

In this paper, we propose to use criteria chains to improve the quality of predicted multi-criteria ratings, in order to further improve the performance of multi-criteria item recommendations. More specifically, we propose the CCA and CCC approaches which use information gain to produce the sequence of criteria to formulate the criteria chains. We predict the rating in each criterion one by one by using the previous predicted multi-criteria ratings as contextual situations. Our experimental results based on the TripAdvisor and YahooMovies demonstrate that criteria chains is able to improve the quality of predicted multi-criteria ratings and further improve the multi-criteria item recommendations. Particularly, CCC which views all of the predicted criteria ratings as contextual situations is able to outperform the CCA which aggregates the predicted multi-criteria ratings in a hybrid of user-specific and item-specific SVR models.

Criteria chains, as a general approach to incorporate criteria dependencies, allow any context-aware recommendation algorithms [3, 17] to be adopted. In addition, the process of predicting criteria ratings can be considered a problem of context suggestion [16, 14, 18], since we view the predicted criteria preference as context information. We will explore the approaches in context suggestion to assist criteria chains in out future work.

REFERENCES

1. Gediminas Adomavicius and YoungOk Kwon. 2007. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems* 22, 3 (2007), 48–55.
2. Gediminas Adomavicius, Nikos Manouselis, and YoungOk Kwon. 2011. Multi-criteria recommender systems. In *Recommender systems handbook*. Springer, 769–803.
3. Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 301–304.
4. Dietmar Jannach, Markus Zanker, and Matthias Fuchs. 2014. Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations. *Information Technology & Tourism* 14, 2 (2014), 119–149.
5. Tanisha Jhalani, Vibhor Kant, and Pragya Dwivedi. 2016. A Linear Regression Approach to Multi-criteria Recommender System. In *International Conference on Data Mining and Big Data*. Springer, 235–243.
6. Yehuda Koren, Robert Bell, Chris Volinsky, and others. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
7. Qiudan Li, Chunheng Wang, and Guanggang Geng. 2008. Improving personalized services in mobile commerce by a novel multicriteria rating approach. In *Proceedings of the 17th ACM conference on World Wide Web*. 1235–1236.
8. Nikos Manouselis and Constantina Costopoulou. 2007a. Analysis and classification of multi-criteria recommender systems. *World Wide Web* 10, 4 (2007), 415–441.
9. Nikos Manouselis and Constantina Costopoulou. 2007b. Experimental analysis of design choices in multiattribute utility collaborative filtering. *International Journal of Pattern Recognition and Artificial Intelligence* 21, 02 (2007), 311–331.
10. Pitaya Poompuang and Wichian Premchaiswadi. 2010. 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*. IEEE, 20–25.
11. Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine learning* 85, 3 (2011), 333–359.
12. Nachiketa Sahoo, Ramayya Krishnan, George Duncan, and Jamie Callan. 2012. Research note-the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! movies. *Information Systems Research* 23, 1 (2012), 231–246.
13. Alex Smola and Vladimir Vapnik. 1997. Support vector regression machines. *Advances in neural information processing systems* 9 (1997), 155–161.
14. Yong Zheng. 2015. Context suggestion: Solutions and challenges. In *2015 IEEE International Conference on Data Mining Workshop*. IEEE, 1602–1603.
15. Yong Zheng. 2017. Situation-Aware Multi-Criteria Recommender System: Using Criteria Preferences as Contexts. In *Proceedings of the 32nd Annual ACM Symposium on Applied Computing*. ACM.
16. Yong Zheng, Bamshad Mobasher, and Robin Burke. 2014. Context recommendation using multi-label classification. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, Vol. 2. IEEE, 288–295.
17. Yong Zheng, Bamshad Mobasher, and Robin Burke. 2015. Integrating context similarity with sparse linear recommendation model. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 370–376.
18. Yong Zheng, Bamshad Mobasher, and Robin Burke. 2016. User-Oriented Context Suggestion. In *Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, 249–258.