Indirect Context Suggestion

Yong Zheng
School of Applied Technology
Illinois Institute of Technology
Chicago, Illinois 60616
yzheng66@iit.edu

ABSTRACT

Context suggestion refers to the task of recommending appropriate contexts to the users to improve the user experience. The suggested contexts could be time, location, companion, category, and so forth. In this paper, we particularly focus on the task of suggesting appropriate contexts to a user on a specific item. We evaluate the indirect context suggestion approaches over a movie data collected from user surveys, in comparison with direct context prediction approaches. Our experimental results reveal that indirect context suggestion is better and tensor factorization is generally the best way to suggest contexts to a user when given an item.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

context, context suggestion, recommender systems

1 INTRODUCTION

Context-aware recommender systems (CARS) try to take context information (e.g., time, location, companion, etc) to estimate user preferences using a "multidimensional" rating function R: Users × Items × Contexts → Ratings [1]. For example, CARS may recommend appropriate movies to us by taking *When* (e.g., weekend or weekday), *Where* (e.g., at home or at cinema) and *with Whom* (e.g., parent, kids, partner) into account. The definition of context may vary from domains to domains, while user's internal status, such as emotional states can also be viewed as context information [6].

The development of CARS left a question – whether we can recommend contexts by going beyond the traditional item recommendations. Recently, a new recommendation task, "context suggestion" [2, 5, 10, 15], is proposed and researchers have demonstrated how promising it is. The idea behind context suggestion is that a list of good item recommendations is not enough to guarantee wonderful user experience. In addition, the appropriate contexts (e.g., time, location, companion, weather, etc) should be suggested to tune up user experiences. For example, taking kids as companions for a couple's romantic vacation may ruin the whole trip sometimes. And, watching a movie at cinema may bring better user experience than enjoying it at home.

We collect user's preferences on movies based on different time, location and companion settings from a survey. We additionally acquire what are the appropriate contexts for the subjects to watch a specific movie. In this paper, we empirically compare different solutions to suggest contexts to a user when given a movie.

UMAP '17, July 9-12, 2017, Bratislava, Slovakia

978-1-4503-4635-1/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3079628.3079654

2 PROBLEMS AND SOLUTIONS

We present the problem statement and discuss different context suggestion solutions in this section.

Table 1: Example: a Context-aware Rating Data Set

User	Movie	Rating	Time	Location	Companion
U1	T1	5	Weekday	Home	Kids
U1	T1	3	Weekend	Cinema	Family
U2	T2	3	Weekday	Cinema	Partner
U2	T3	4	Weekday	Home	Kids
U3	T4	2	Weekend	Home	Partner

Table 1 shows an example of context-aware movie data which contains five rating profiles given by three users on four movies in different contextual situations. In our discussions, we will use the term *contextual dimension* to denote the contextual variable, such as "Location", "Time" and "Companion". The term *contextual condition* refers to a specific value in a contextual dimension, e.g. "Home" and "Cinema" are two contextual conditions for the dimension "Location". *Context* or *contextual situation* therefore refers to a combination of contextual conditions, e.g., {Weekday, Home, Kids}.

In this paper, the problem we are interested in is that how to suggest the appropriate contexts for a user to better enjoy a given item. Our previous research [10] introduces two types of possible solutions that can be summarized as follows.

2.1 Direct Context Prediction

By this way, the context suggestion task is transformed to multiple context prediction task. And multi-label classification (MLC) can be applied to predict which context conditions should be recommended. More specifically, each context condition is viewed as a single label in the MLC task, and the context prediction process can be realized by binary classification to predict whether each context condition is appropriate or not. In [10], we discover that Classifier Chains (CC) and Label Powerset (LP) are two most effective solutions in MLC.

2.2 Indirect Context Suggestion

Indirect context suggestion, by contrast, is going to predict a score for each context conditions and rank them by the predicted score to produce top-N recommendations.

In CARS research, we try to exploit the multi-dimensional rating space to recommend a list of appropriate items by taking a user and contexts as inputs. From another perspective, we can also suggest appropriate contexts to a user by taking a user and an item as the input. In this paper, we try to explore the indirect context suggestion by reusing the context-aware recommendation algorithms.

3 EXPERIMENT AND RESULTS

To apply the indirect context suggestion, we need a data set with user's contextual ratings on the items and the user's tastes on contexts. There are no such data available. The data that was used in the previous research in [2] is a data set without user's contextual ratings.

In this case, we conduct a user survey and finally obtain the data with 5043 movie ratings given by 97 users on 79 movies. On average, each user gave 51 ratings on different movies in various context situations. In addition, we ask the subjects to explicitly indicate the appropriate contexts to watch selected movies and what are the general contexts they prefer to watch a movie. We perform the experimental evaluations based on a 5-fold cross validation by using normalized discounted cumulative gain (NDCG) as the metric.

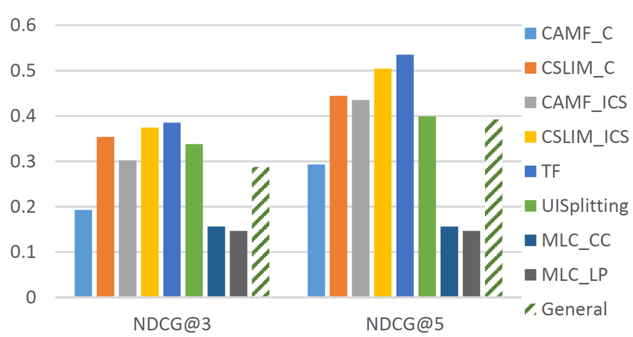

Figure 1: Experimental Results Based on NDCG

The context-aware recommendation algorithms we used for indirection context suggestion are: UISplitting [9], CAMF_C [3], CSLIM_C [11], CAMF_ICS [14], CSLIM_ICS [13] and Tensor Factorization (TF) [4], where we use CARSKit [12] to produce the results. The experimental results based on top-3 and top5 context suggestion can be shown in Figure 1.

We can observe that only the UISplitting, TF and CLSIM algorithms can beat the user's general preferences on contexts in this data, while TF is the best performing method. The results also reveal that the indirect context suggestion methods can outperform the direct context prediction approaches (indicated by MLC_CC and MLC_LP).

4 CONCLUSIONS

In this paper, we collect contextual ratings and user's preferences on contexts from user surveys. We empirically compare the indirect context suggestion with the direct context prediction approaches. We find indirect context suggestion is better and TF becomes the best performing technique in our experiments. We plan to create real-world applications to collect actual data in our future work to boost the development and research on context suggestions.

Our latest research found that criteria can also be viewed as contexts [7, 8]. It implies that we can also suggest selected criteria to the end users. For example, we may recommend "clean room" or "convenient location" to the hotel booking applications like tripadivsor.com.

REFERENCES

[1] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alexander Tuzhilin. 2011. Context-Aware Recommender Systems. *AI Magazine* 32, 3 (2011), 67–80.

[2] Linas Baltrunas, Marius Kaminskas, Francesco Ricci, Rokach Lior, Bracha Shapira, and Karl-Heinz Luke. 2010. Best usage context predictions for music tracks. In *The 2nd Workshop on Context-aware Recommender Systems*.

[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] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 79–86.

[5] Yong Zheng. 2015. Context Suggestion: Solutions and Challenges. In *Proceedings of the 15th IEEE International Conference on Data Mining Workshops*. IEEE, 1602–1603.

[6] Yong Zheng. 2015. A revisit to the identification of contexts in recommender systems. In *Proceedings of the Conference on Intelligent User Interfaces Companion*. ACM, 133–136.

[7] Yong Zheng. 2017. Criteria Chains: A Novel Multi-Criteria Recommendation Approach. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 29–33.

[8] Yong Zheng. 2017. Situation-Aware Multi-Criteria Recommender System: Using Criteria Preferences as Contexts. In *Proceedings of the ACM Symposium on Applied Computing*. ACM, 689–692.

[9] Yong Zheng, Robin Burke, and Bamshad Mobasher. 2014. Splitting approaches for context-aware recommendation: An empirical study. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, 274–279.

[10] Y. Zheng, B. Mobasher, and R. Burke. 2014. Context Recommendation Using Multi-label Classification. In *Proceedings of the 13th IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE, 288–295.

[11] Y. Zheng, B. Mobasher, and R. Burke. 2014. CSLIM: Contextual SLIM Recommendation Algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, 301–304.

[12] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2015. CARSKit: A java-based context-aware recommendation engine. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE, 1668–1671.

[13] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2015. Integrating Context Similarity with Sparse Linear Recommendation Model. In *Proceedings of the Conference on User Modeling Adaptation and Personalization*. 370–376.

[14] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2015. Similarity-Based Context-aware Recommendation. In *Proceedings of the 2015 Conference on Web Information Systems Engineering*. Springer Berlin Heidelberg, 431–447.

[15] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2016. User-Oriented Context Suggestion. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 249–258.