An Evaluation Framework for Interactive Recommender Systems

Oznur Alkan
oalkan2@ie.ibm.com
IBM Research
Dublin, Ireland

Elizabeth M. Daly
elizabeth.daly@ie.ibm.com
IBM Research
Dublin, Ireland

Adi Botea
adibotea@ie.ibm.com
IBM Research
Dublin, Ireland

ABSTRACT

Traditional recommender systems present a relatively static list of recommendations to a user where the feedback is typically limited to an accept/reject or a rating model. However, these simple modes of feedback may only provide limited insights as to why a user likes or dislikes an item and what aspects of the item the user has considered. Interactive recommender systems present an opportunity to engage the user in the process by allowing them to interact with the recommendations, provide feedback and impact the results in real-time. Evaluation of the impact of the user interaction typically requires an extensive user study which is time consuming and gives researchers limited opportunities to tune their solutions without having to conduct multiple rounds of user feedback. Additionally, user experience and design aspects can have a significant impact on the user feedback which may result in not necessarily assessing the quality of some of the underlying algorithmic decisions in the overall solution. As a result, we present an evaluation framework which aims to simulate the users interacting with the recommender. We formulate metrics to evaluate the quality of the interactive recommenders which are outputted by the framework once simulation is completed. While simulation alone is not sufficient to evaluate a complete solution, the results can be useful to help researchers tune their solution before moving to the user study stage.

ACM Reference Format:
Oznur Alkan, Elizabeth M. Daly, and Adi Botea. 2019. An Evaluation Framework for Interactive Recommender Systems. In *27th Conference on User Modeling, Adaptation and Personalization Adjunct (UMAP'19 Adjunct), June 9–12, 2019, Larnaca, Cyprus.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3314183.3323680

1 INTRODUCTION

Interactive recommender systems have gained attention by allowing users to have more control and making them more actively involved in the process of their recommendations. One challenge that has made it difficult for researchers to progress in this field is that, there exists no standard evaluation mechanism for such a framework. As a result, we have developed a sound evaluation framework to simulate users as if they are interacting with the recommender. The interaction we designed assumed to be performed via conversation. The system can work with any recommender algorithm that supports recommendations based on a user profile and adapting the recommendations based on the learnt preferences. In addition, we formulated metrics that combine measures from two related areas of research, *recommendation systems* and *dialogue management*, in order to evaluate the interactive recommender from both recommender accuracy and dialogue quality perspectives.

The interactions considered during designing the framework are as follows:

Item feedback. A user can accept or reject an item.

Item exploration, explanations and correcting incorrect assumptions. A user can examine item details or ask for explanation for why an item is recommended. Once a user understands the motivation behind a recommendation, then the user has the opportunity to correct any incorrect assumptions.

Providing explicit preference. A user can provide preferences (negative/positive/complex) towards features of items at any point of conversation. The preference can be provided as a positive (*e.g. I like horror movies.*) or negative (*e.g. I do not like romantic comedies.*) preference towards a single feature, or a compound feature including both a positive and a negative preference, (*e.g. I want to watch a movie of Keanu Reeves, but not a romantic one.*)

Preference elicitation. The system can provide the user with a choice of features to provide feedback on, to enrich the user preference profile (*e.g. Do you prefer comedy movies?*). The preference elicitation implemented uses a feature selection method based on *Information Gain* [2] to select the features that can split the recommendation space better and allow the recommender to filter out candidates. The output of the preference elicitation process is therefore the feature whose value will be asked to the user.

Although the framework currently only supports the above interaction mechanisms, the solution is generic enough to support extensions to other interaction mechanisms as needed.

2 PROPOSED EVALUATION FRAMEWORK

In order to run the evaluation framework, any recommendation dataset that includes user-item ratings together with the item related data can be used. The data for each user is split into training and test, where the training items are used to generate the first set of recommendations and the test items, which is called as the *look-ahead data*, is used to simulate the interaction.

Each interaction is captured by a conversation state denoted by the nodes in Figure 1. The *look-ahead data* is used in order to determine the simulated users' choices at each state. The child nodes of each parent node reflect the possible next states that are

UMAP'19 Adjunct, June 9–12, 2019, Larnaca, Cyprus
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6711-0/19/06…$15.00
https://doi.org/10.1145/3314183.3323680

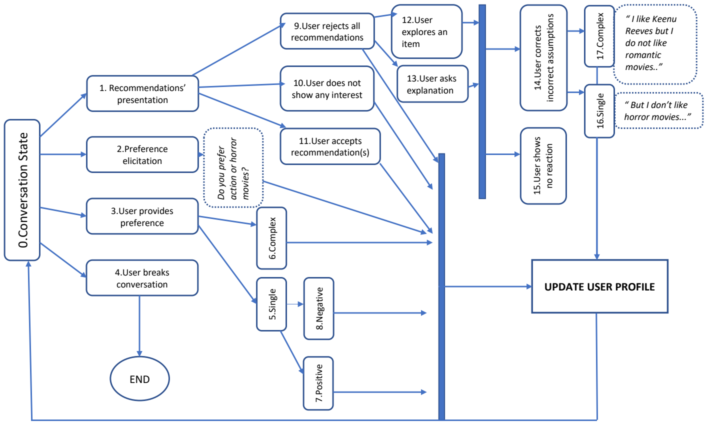
0.Conversation State
1. Recommendations’ presentation
2.Preference elicitation
3.User provides preference
4.User breaks conversation
END
Do you prefer action or horror movies?
5.Single
6.Complex
7.Positive
8.Negative
9.User rejects all recommendations
10.User does not show any interest
11.User accepts recommendation(s)
12.User explores an item
13.User asks explanation
14.User corrects incorrect assumptions
15.User shows no reaction
16.Single
17.Complex
“ I like Keenu Reeves but I do not like romantic movies..”
“ But I don’t like horror movies…”
UPDATE USER PROFILE

Figure 1: User Simulation Outline

chosen either by randomly or by leveraging the *look-ahead data*. For example, when items are recommended, users may reject all of them or they may want to explore one of the recommended items to observe the properties of it and seek an explanation for why that item was recommended.

The goal is to simulate user interaction until the user breaks the conversation. Figure 1 shows the currently supported conversational states.

Interaction initialization state *(state 0):* At this point there are *4* possible interactions; recommendations’ presentation *(state 1)*, system asking a preference elicitation question *(state 2)*, user providing explicit preference *(state 3)*, and user breaking the conversation which ends the whole evaluation dialogue with a test user *(state 4)*. Next state is chosen in a weighted random manner.

Recommendations’ presentation *(state 1)*: The *look-ahead data* is used to assess the user’s reaction to the recommendations. If the user has negatively rated the presented items, then *state 9* is selected. If the future items in the *look-ahead data* have no rating for the presented items, then we infer that the user has no interest for the suggestions *(state 10)*, whereas if the user has positively rated at least one recommended item, *state 11* is selected.

Preference elicitation *(state 2):* The system generates a preference elicitation option for the user, and the option selected is determined based on the *look-ahead data*, where the features associated with future items are used to infer the preferences of the user.

User providing preference *(state 3)*: In a similar manner to above, the *look-ahead data* is used to examine the features of the future items, where we can generate a preference the user chooses to give to the system.

User rejects all recommendations*(state 9)*: If there are no successful recommendations, we assume that the user takes an action to provide further information by either exploring the recommendations *(state 12)* or seeking an explanation *(state 13)*. We randomly select between these two actions, where the user can either give feedback on the features associated with an item recommended, or a feature that appears in the recommendation’s explanation. For example, recommender may explain recommending the movie *November Rain* by saying *"I recommend you November Rain because you like Keenu Reeves and romantic movies"*. User may correct the recommender by responding as *"I like Keenu Reeves but I dont like romantic movies"*. The feedback user provides is inferred based on the *look-ahead data* in order to determine positive and negative features associated with the future items.

2.1 Metrics

Typical recommender system metrics focus on the accuracy of a presented list, however, considering interactions with the user, we want to be able to capture what the overhead is on the user. For example, simply rating movies might take quite a number of recommendations to hit a successfull recommendation, whereas asking the user if they prefer *drama* or *horror* might lead to success with a smaller number of interactions. As a result, we designed a number of metrics to evaluate not only the *accuracy* of the recommendations but also the amount of additional interactions between the user and the recommender needed, which reflects the quality of the interactions.

DialogTurn (DT): #dialogue turns, where each utterance of the user and the recommender is counted as one turn.

DialogSuccessRate (DSR): avg #successfull recommendations presented to the user during per dialogue turn.

AP@k: average precision of top *k* recommendations [1], used in calculation of *AP@kDT*.

AP@kDT: avg normalized AP@k by the #dialogue turns.

We integrated these metrics within the framework, and at the end of the evaluation with each test user, the average of the calculated metrics over test users is presented.

3 CONCLUSION

Evaluating the impact of the user interactions in interactive recommendation systems typically requires an extensive user study which is challenging. In this paper, we present an evaluation framework and metrics to evaluate the quality of the interactive recommenders, which can help researchers tune their solution before moving to the user study stage.

REFERENCES

[1] Hang Li. 2011. A Short Introduction to Learning to Rank. 94-D (10 2011), 1854–1862.

[2] Yiming Yang and Jan O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML âĂŽ97)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 412–420. http://dl.acm.org/citation.cfm?id=645526.657137