History-Aware Critiquing-Based Conversational Recommendation

Yasser Salem
School of Electronics,
Electrical Engineering
and Computer Science
Queen's University Belfast
Belfast, BT7 1NN, UK
ysalem01@qub.ac.uk

Jun Hong
School of Electronics,
Electrical Engineering
and Computer Science
Queen's University Belfast
Belfast, BT7 1NN, UK
j.hong@qub.ac.uk

ABSTRACT

In this paper we present a new approach to critiquing-based conversational recommendation, which we call History-Aware Critiquing (HAC). It takes a case-based reasoning approach by reusing relevant recommendation sessions of past users to short-cut the recommendation session of the current user. It selects relevant recommendation sessions from a case base that contains the successful recommendation sessions of past users. A past recommendation session can be selected if it contains similar recommended items to the ones in the current session and its critiques sufficiently overlap with the critiques so far in the current session. HAC extends experience-based critiquing (EBC).

Our experimental results show that, in terms of recommendation efficiency, while EBC performs better than standard critiquing (STD), it does not perform as well as more recent techniques such as incremental critiquing (IC), whereas HAC achieves better recommendation efficiency over both STD and IC.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrievals**]: On-line Information Services—*Web-based services*

Keywords

Conversational Recommendation, Recommender Systems, Experience-Based Critiquing

1. INTRODUCTION

Recommender systems are an important part of the online world and especially critical in an e-commerce setting. They provide users with a proactive and personalized information discovery service. A recent approach [2] to improving the efficiency of critiquing-based recommender systems has explored the idea that the critiquing histories or experiences of past users carry important information about feature preferences and trade-offs. The approach is called *experience-based critiquing* (EBC), which improves *standard critiquing* (STD) [1] by retrieving the recommendation sessions of past users, selecting relevant sessions that have similar critiques to the ones in the current session and using the successfully accepted recommendations in the relevant sessions as a source of candidate recommendations. It has been shown in [2] that EBC achieves significant reduction in session length over STD. This paper continues this line of research. Our starting point is to consider the application of EBC to more recent critiquing approaches such as *incremental critiquing* (IC) [3]. IC achieves significant reduction in session length over STD, by harnessing a profile of a user's previous critiques in a recommendation session and using the profile to influence the selection of the next recommended item in the session. Since sessions are already much shorter in IC than in STD, there is less chance for EBC to shorten them further. In fact we have found that EBC does not have beneficial impact on IC and it tends to produce much longer sessions than IC. In this paper, we propose a new approach which extends EBC in an effort to improve its efficiency over both STD and IC.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

2. HISTORY-AWARE CRITIQUING

EBC looks for similarity across critique patterns. However this sometimes results in items being recommended which the user may find unexpected. For example, suppose that the user of a restaurant recommender system is recommended an Indian restaurant and then critiques the recommendation by asking for a cheaper price. At this point, EBC compares the user's critiques so far with the critiques in each of the experience cases in the case base in order to identify a set of relevant sessions and generate a new recommendation. The user may be presented with a recommendation which satisfies their critiques but it may not meet their expectations. For example, the new recommendation may be a cheaper French restaurant. In other words, EBC focuses only on the critique patterns and does not pay attention to the types of items that the user has been recommended so far, at least not directly. As a result, new items may be recommended that may appear to be quite different on a feature by feature basis from the items that the user has been recommended previously. This problem is the motivation for this work. Since the past recommendation sessions contain both recommendations and critiques, we propose to use the knowledge from both to generate new recommendations. To do so we add a new component to EBC to allow us to take into account item similarity when selecting relevant recommendation sessions. In a typical critiquing-based recommendation session, the user starts with high-level understanding

of their own needs. During the course of the recommendation session this is refined, as the user critiques various features of the recommended items. Each recommendation session, s, is represented as a sequence of *recommendation-critique pairs*, $s = \{p_1, p_2, ..., p_n\}$, where $p_i = (r_i, c_i)$, for $i = 1, 2, ..., n$, with r_i representing a recommendation and c_i representing the critique applied to r_i.
Furthermore, c_i is represented as a triple, $(f_i, v_i, type_i)$, where f_i refers to a feature in r_i, v_i is the value of f_i, and $type_i$ is the type of c_i (typically, $type_i \in \{<, >, =, <>\}$). We assume that each session terminates when the user chooses to *accept* the current recommendation, indicating that they are satisfied with it, or when they choose to *stop* the session, presumably because they have grown frustrated with the recommendations received so far. When a user applies a critique c_m to a recommended item r_m, the user's current sequence of recommendation-critique pairs, $p_1, p_2, ..., p_m$, is used as a query, q_T, over the case base of past recommendation sessions, in order to identify a set of *relevant sessions*.

$$averageSimilarity(q_T, s) = \frac{\sum_{r_i \in q_T} \sum_{r_j \in s} sim(r_i, r_j)}{|q_T| * |s|} \quad (1)$$

Equation 1 defines the average similarity between the recommended items in the recommendation session of the current user, q_T, and the ones in a past recommendation session, s, from the case base. The $sim(r_i, r_j)$ function in Equation 1 calculates similarity between two recommended items, r_i and r_j, which are a recommended item in the current session and a recommended item in a past session respectively.

$$Quality(q_T, s) = \alpha * averageSimilarity(q_T, s) + (1 - \alpha) * OverlapScore(q_T, s) \quad (2)$$

We introduce a quality metric as defined in Equation 2 that combines item similarity as defined in Equation 1 and critique overlap, where the relative weights of item similarity and critique overlap are controlled by weighting factor α.

$$S^{REL} = RelevantSessions(q_T, S) = \left\{ s \in S : Quality(q_T, s) > t \right\} \quad (3)$$

Equation 3 defines how the quality metric is used to identify a set of relevant recommendation sessions from a set of past sessions, S, in the case base. The minimum quality required is controlled with a threshold, t. Before the evaluation, we ran simulations to set the values of t and α. This simulation revealed that their values are set to 0.8 and 0.75 respectively. Future work will further investigate how to choose the best values for both t and α. In a similar manner to EBC, Equation 4 defines how the accepted recommendation from the relevant past recommendation session with the maximum quality score is selected as the new recommendation, r_T, in the current recommendation session.

$$r_T = Recommend(S^{REL}) = \underset{\forall s \in S^{REL}}{argmax} \left(Quality(q_T, s) \right) \quad (4)$$

3. EXPERIMENTAL EVALUATION

In this section we present an experimental evaluation of HAC in comparison with EBC, and with both STD and IC. We used the same datasets as used in [2] (see [2] for full details).

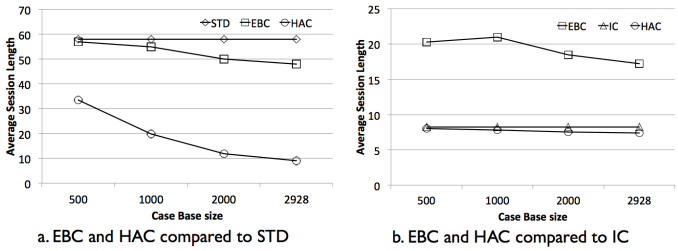

Figure 1: EBC and HAC compared to STD and IC

3.1 Algorithms & Methodology

For the purpose of the evaluation we consider two families of algorithms for 6 different recommendation techniques. The *standard critiquing family* includes STD, EBC in conjunction with STD, and HAC in conjunction with STD. The *incremental critiquing family* includes IC, EBC in conjunction with IC, and HAC in conjunction with IC.

3.2 Results

In order to compare the performance of HAC to EBC and the baseline techniques (STD and IC), we performed two sets of tests. In the first set we compare HAC and EBC with STD. In this instance, HAC and EBC both revert back to STD when a relevant past recommendation session cannot be found. In the second set we compare HAC and EBC with IC, and both HAC and EBC revert back to IC when a relevant past recommendation session cannot be found. Figure 1a. shows the average session lengths of both EBC and HAC with reverting back to STD. HAC achieves much more reduction in session length: for the largest case base size, the average session length of HAC is reduced to under 10 cycles (compared to 58 for STD and 48 for EBC). When comparisons are made with both EBC and HAC reverting back to IC, the benefits of HAC can still clearly be seen, as shown in Figure 1b. HAC achieves better performance than EBC and outperforms IC. Its performance further improves as the size of the case base increases.

4. CONCLUSIONS

We have presented HAC as an extension to EBC by introducing item similarity explicitly into the recommendation process. The results of our large-scale experiments are promising. They show that HAC achieves marked improvements over EBC and more modest improvements over IC.

5. REFERENCES

[1] R. Burke, K. Hammond, and B. Young. The FindMe Approach to Assisted Browsing. *IEEE Expert*, 12(4):32–40, 1997.

[2] K. McCarthy, Y. Salem, and B. Smyth. Experience-Based Critiquing: Reusing Critiquing Experiences to Improve Conversational Recommendation. In *Proceedings of the 18th International Conference on Case-Based Reasoning, (ICCBR-2010)*, pages 480–494, 2010.

[3] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Incremental critiquing. *Knowledge-Based Systems*, 18(4-5):143–151, 2005.