International Workshop on Benchmarking Adaptive Retrieval and Recommender Systems (BARS)

held in conjunction with
The 36th Annual ACM SIGIR
Conference

Dublin, Ireland
1 August 2013

Proceedings

Organizers
Pablo Castells – UAM, ES
Frank Hopfgartner – TU Berlin, DE
Mounia Lalmas – Yahoo! Labs, ES
Alan Said – CWI, NL

Program Committee
Dyaa Albakour – University of Glasgow, UK
Leif Azzopardi – University of Glasgow, UK
Alejandro Bellogin – CWI, NL
Nicholas Belkin – Rutgers University, USA
Toine Bogers – RSLIS, DK
Pia Borlund – RSLIS, DK
Paolo Cremonesi – Politecnico di Milano, IT
Noriko Kando – National Institute of Informatics, JP
Alexandros Karatzoglou – Telefónica, ES
Bart Knijnenburg – UC Irvine, USA
Udo Kruschwitz – University of Essex, UK
Nikos Manouselis – Agroknow, GR
Martha Larson – TU Delft, NL
Neal Lathia – University of Cambridge, UK
Denis Parra – University of Pittsburgh, USA
Ian Ruthven – University of Strathclyde, UK
Domonkos Tikk – Gravity R&D, HU
Ryen White – Microsoft, USA
Michelle Zhou – IBM, USA

Additional Reviewer
Benjamin Kille – TU Berlin, DE

Keynote Speaker
Torben Brodt – plista GmbH, DE

Panel Members
Paul N. Bennett – Microsoft Research, USA
Jeremy Pickens – Catalyst Repository Systems Inc, USA
Jimmy Lin – University of Maryland, USA
Neil Hurley – University College Dublin, IE

The proceedings are available for download at
`http://bars-workshop.org/`.

Table of Contents

Program

09:00 – 09:15	*Welcome*	
09:15 – 10:15	*Invited Talk*	
	T. Brodt (plista GmbH, Germany)	*The Search for the Best Live Recommender System*
10:15 – 11:00	*Coffee break*	
11:00 – 12:00	*Oral Presentations*	
	D. Wilson and N. Najjar (University of North Carolina at Charlotte, USA)	*Tradeoffs in Evaluation Strategies for Group Recommender Systems*
	S. Gupta and S. Chakraborti (IIT Madras, India)	*Evaluating Conversational Recommender Systems based on Preference Based Feedback*
12:00 – 12:40	*Panel session*	

Invited Keynote

The Search for the Best Live Recommender System

Torben Brodt
plista GmbH
Berlin, Germany
tb@plista.com

ABSTRACT

plista is a data-driven content- and ad distribution network, mainly active in the German speaking world. Its technology is used on thousands of premium websites for recommending millions of news articles and ads to even more users. Currently, over 5,000 requests are processed per second. In this talk, I will present challenges and opportunities that arise from handling such vast amounts of data in real-time. In particular, I will illustrate why users' context (e.g., category, geo-location, day of the week) plays a key role in the provision of successful recommender algorithms and will outline the importance of a thorough evaluation of our techniques. Besides, I will talk about our experience in organizing a real-time news recommender contest which provides researchers the chance to run their algorithms in our production system, thus benefiting from real feedback from real users.

Papers

Tradeoffs in Evaluation Strategies for Group Recommender Systems

David C. Wilson
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC, USA
davils@uncc.edu

Nadia A. Najjar
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC, USA
nanajjar@uncc.edu

ABSTRACT

Making recommendations to groups of users gives rise to significant challenges for group modeling and recommendation, but also particularly for evaluating group recommender systems. A common approach for scalability in group recommender research is to generate synthetic groups from traditional, single-user datasets, and we focus on this context. In evaluating recommendations for synthetic groups, the "actual" group preference for an item — the baseline for measuring recommendation accuracy — has typically been modeled as average rating across group members. However, there is comparatively little evidence that real group decisions rely almost solely on an average strategy. As a result, such evaluations (and the system development that relies on them) may not always provide the best model to measure group recommender behavior. To address this issue, we are investigating alternative ways to model "actual" group preferences and the implications for evaluation in the fit of group recommendation strategy to the choice of how the group evaluation baseline is modeled. To help understand potential tradeoffs in the evaluation space, this paper compares several models for the "actual" group evaluation baseline, including Average, Least Misery and Most Happiness. Results show that the commonly used Average model can overestimate reported accuracy results and suggest factors for consideration in evaluation design for group recommendation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Algorithms, Experimentation, Standardization

BARS'13, August 1, 2013, Dublin, Ireland.

Keywords

Evaluation, Group recommendation, Collaborative filtering, Memory-based

1. INTRODUCTION

Traditional recommender systems focus on an individual end user as the target for making recommendations. More recently, however, the issue of group recommendation has received increasing attention [12, 3]. Group recommender systems balance individual preferences across a group of users, in order to make a recommendation for the group as a whole. Such recommendations typically address domains with social aspects and shared-consumption needs, such as finding a movie to watch together [18, 8, 24], finding a restaurant [15] or recipe [4] for eating together, or finding a place to travel together [16, 2, 11].

Research in group-based recommender systems has been focused on approaches to model the group for recommendation and two main approaches have been proposed: aggregating preferences and aggregating recommendations [12]. These strategies utilize recommendation approaches validated for individual users, and so the aggregation strategy is typically the defining feature for work in group-based recommenders. Group modeling strategies are inspired by Social Choice Theory and center around modeling the achievement of consensus among the group [13]. Variations have also been investigated that consider personalities of and social interactions among group members [7, 21].

To assess the quality of individual user recommendations, researchers commonly utilize offline evaluations that employ readily available, substantial data sets (e.g., Netflix prize[1], MovieLens[2]). This approach can be used to repeatedly conduct large scale evaluations of any proposed technique. However, when it comes to group-based recommender systems such datasets are not readily available. Generating group-based data directly requires extra overhead in recruiting the groups together and getting them to cooperate and interact towards a common goal at the same time. To address scalability in evaluation, researchers have been utilizing synthetic groups, generated from single-user data sets, to evaluate various approaches to group recommendations [23, 3, 1, 6, 5]. The problem with this approach lies in the scope of establishing the baseline "actual" group preference for an item

[1] www.netflixprize.com
[2] www.movielens.org

— the baseline for measuring recommendation accuracy for these synthesized groups.

Traditional leave-one-out style evaluation for recommender accuracy follows a straightforward baseline process:

1. Select a user for testing.
2. Select an item for testing.
3. Predict the selected user's rating for the selected item using the recommender approach being evaluated.
4. Compare the predicted value to the actual item rating in order to find the magnitude of difference.

In step 4, the process is straightforward, because the actual rating is known and fixed.

The same general approach can be applied to leave-one-out style evaluation for group recommender accuracy:

1. Select a group for testing.
2. Select an item for testing.
3. Predict the selected group's rating for the selected item using the group. recommender approach being evaluated
4. Compare the predicted value to the group's actual rating in order to find the magnitude of the difference.

In the group case, however, the final step can be problematic, as the group's actual overall rating is typically not known. Thus the "actual" group preference itself must be modeled in some way. Moreover, we note that this is true both for synthetic group studies based on individual user data, as well as user studies with real groups of people [1, 21]. Modeling the ground truth group preference to establish the evaluative baseline, in addition to modeling the recommendation prediction itself, has the potential to decrease precision in overall evaluation results. We believe that this is a significant issue for evaluation in group recommender systems, and our research focuses on investigating how to characterize the impact and tradeoffs in such evaluations.

In [17] we identified two distinct points where the need for group modeling arises in group recommender systems evaluation. We refer to the first as the *Recommendation Group Preference Model* (*RGPM*) and it is how we model a group for the purpose of making recommendations (i.e., what a group's preference outcome *will be*). The second is the *Actual Group Preference Model* (*AGPM*) which is how we determine an "actual" group preference, based on outcomes in user data, in order to represent ground truth for evaluation purposes (i.e., what a group's preference outcome *was*).

In evaluating recommendations for synthetic groups, the *AGPM* — the baseline for measuring recommendation accuracy — has typically been modeled as an average rating across group members (e.g., [3]). However, there is comparatively little evidence that real group decisions rely almost solely on an average strategy. For example, [13] and [12] note that different groups employ a variety of different strategies. As a result, evaluations that consider only an average strategy *AGPM* (and the system development that relies on them) may not always provide the best model for group recommender behavior. Recent work by Quijano-Sanchez et. al. [20] utilized human subjects to establish ground truth for the synthesized groups used in their evaluation of a case-based aggregation model for group recommenders. They based the *AGPM* on a voting scheme of a panel of experts rather than a model based on the individual known preferences of the group members.

In the absence of one true *AGPM*, the meta-issue arises of how to evaluate candidate evaluation models. In order to do so, there are essentially two traditional options: finding a baseline context for comparing *AGPMs*, or relative performance comparison among different evaluation models. Keeping in mind that the the dependent variable in such experiments is a component of the evaluation strategy itself. This issue of how to model the *AGPM* in the evaluation of group recommenders using synthesized groups motivated us to investigate alternative approaches for *AGPMs*. Of course changing the evaluation baseline will show differences in outcomes, but we are interested in investigating the shape and extent of such differences as a way to inform the process of evaluation. We examine the scope of potential tradeoffs in the choice of an aggregation model for recommendations for groups when actual group preference may vary. In this paper we lay out a brief survey of evaluation techniques in this context. This can provide group recommender system developers with a better understanding of the implications in choosing a particular *AGPM* baseline.

2. RELATED WORK

Evaluations in group based recommender systems, whether utilizing user studies or synthetic groups, had to make concessions in the evaluation set up when evaluating the performance of models of the decision making of groups. For example, some studies involved individual subjects that were asked to make decisions for a group and that was used to model the actual group preference [13]. On the other hand evaluations with synthetic groups evaluated performance against the individual group members' preferences rather an actual group preference [3].

2.1 Evaluation with User Studies

In PolyLens [18], one of the earliest group recommenders, they used a survey to gauge users' satisfaction rather than an accuracy measure (e.g., RMSE) of the performance of their group recommender system. They conducted a field study of an online movie recommender system where users were able to form groups and receive recommendations tailored to the group. The evaluation focused on understanding the user experience with the system. They measured how users formed groups, how they used the groups, and how the experience of users who used groups differed from the experience of users who did not use groups. They also asked the users for their opinion about the value of the recommendations. They showed that while users liked and used group recommendation, they disliked the minimize misery strategy and there's a need for better social value group model.

Masthoff [13] employed user studies to determine which group aggregation strategies people actually use. The subjects of this study were not part of a group but rather individuals that were asked to decide for a group given the individual group member's item preferences. The subjects were given the same individual rating sets for a group of three people on a collection of video clips and asked to decide which clips the group should see. They were also asked to explain why they made that selection. Their results indicated that user groups employ a variety of strategies in coming to consensus and not just the average strategy exclusively. In this study, the subjects particularly used the following strategies: Average, Average Without Misery and Least Misery.

Amer-Yahia et al. [1] conducted a user study on Amazon's Mechanical Turk. In considering factors that affect group formation, they formed groups based on size and group cohesiveness. The aim of this evaluation was to understand the impact of the group size in reaching consensus among the group members and how the satisfaction with the group recommendation is affected by the group. After the individual preferences of the users were collected, groups were formed and recommendations for the groups were generated. To obtain ground truth the users were asked to evaluate the group recommendations for the groups to which they belonged, by indicating if each item is suitable for recommendation given the group context. They measured the quality of the results of a ranked list using the normalized discounted cumulative gain (nDCG) for the prediction lists generated for the group. To get a group evaluation result the nDCG measure was computed for each group member and the average, of these individual results, was considered the effectiveness of the group recommendation. We note that, in this study, the evaluators for the group recommendations were complemented with some users that were not actually members of those groups, but were asked to pretend that they were.

Berkovsky and Freyne [4] evaluated the performance of the two approaches to group recommendation (merging profile and merging recommendation) using a recipe recommendation system for groups. They reported better performance in the recipe recommendation domain when aggregating the user profiles rather than aggregating individual user predictions. The evaluation consisted of 108 families, where 70 of them were a family with size one, which is an individual member not a group of users. They evaluated the recommendations against the preferences provided by the individual users and not a group based assessment of the results.

2.2 Evaluation with Synthetic Groups

Quijano-Sanchez et. al. [19] uses synthesized groups as well as human subjects to evaluate a case-based approach to preference aggregation rather than a model based on social functions. They simulated groups from the MovieLens dataset where each group had at least 15 commonly rated items among its group members. These 15 items were used as the test set and recommendations were evaluated based on them. They created a total of 100 groups with different sizes (2,3,4,5,6,7). To establish ground truth they used a panel of 4 experts to decide which movie in the group's test set the group was most likely to agree on. The expert users were presented with all the information about the group members and the real ratings of the 15 test items for each group, and were asked to give an ordered list of the three movies from the group's test set that they thought the group would agree on. A voting scheme was then applied to these lists to establish a final ranked list as the ground truth for each group. To overcome the issue of establishing ground truth for synthesized groups, this work utilized real users to make that judgment. A model is used to aggregate the judgments of these outside users into a group's preference which, in turn, is another variable in the evaluation.

Salamó et. al. [23] evaluated several strategies used to aggregate satisfaction of the individual preferences of group members for an item in the case base. They utilized synthetic groups generated from user profiles of critique preferences from a critiquing-based, travel recommender system (CATS). They had 34 individual profiles which they used to generate groups of various sizes (3, 4, 6, and 8), with 300 groups forming 3 sets of 100 groups, where each set was made up of members with certain similarity characteristics (*similar*, *diverse* and *mixed*). To establish ground truth individual users were asked to indicate the ideal set of ski holiday features to form the "Perfect Product" for each user. For each test group, they generated recommendations using different strategies. They evaluated the prediction accuracy of a single top recommendation and a final recommendation list of 5 products, by comparing the average similarity of the recommendation product to the "Perfect Products" of the group members. Their results indicate that Multiplicative, Borda, and Average strategies perform best across all group sizes. The authors point out that a main challenge in their evaluation is locating a baseline. They compare their strategies against each other and a random baseline of a product chosen randomly from the product space.

Baltrunas et. al. [3] also used simulated groups to evaluate the aggregation strategies of ranked lists. They used the MovieLens data set to simulate groups of different sizes (2, 3, 4, 8) and different degrees of similarity (high and random). They randomly generated 1000 groups for each condition. They measured the effectiveness of the predicted rank list using nDCG for each group member and the average as the group's nDCG. To establish ground truth the dataset was randomly divided into training and testing sets. The nDCG was calculated for each group member based on the items that appeared in that user's testing set. In this approach it is possible that the performance measure is calculated using lists of different lengths and different items among the group members. They reported that varying the group size, the variation of the effectiveness of the group recommendations is not large for groups of size 2, 3, and 4. They also noted that when increasing the group size the effectiveness of the group recommendations tends to decrease only for randomly generated groups. For groups with high inner similarity, as the group size increases the effectiveness increases as well. They didn't assert a clear winner between these strategies since the best performing method in each evaluation depended on the group size and inner group similarity. In this evaluation they did not evaluate the recommendations generated for the group to asses the quality of a group recommendation rather they evaluated them to see if they provided better individual recommendations.

Using a group recommender for tourist activities Garcia et al. [6] evaluated three methods for aggregating individual preferences into a group preference profile. The dataset used was composed of 60 individual user profiles containing general preferences, demographic data, visited places and the user's degree of satisfaction when visiting these places. They randomly generated synthetic groups with sizes varying between 2, 3, 4, 5 and 6. They did not report how many groups were generated. They compared the quality of recommendation lists of 10 items generated using these approaches for the group. They used the average and standard deviation of the utility of a recommendation list over all the group members. Their results show that the utility, on average, is similar across all group sizes.

Amer-Yahia et al. [1] also used simulated groups to measure the performance of different strategies that are based on a top-k Threshold Algorithm. They evaluated their approach using synthetic groups generated from the MovieLens data where groups were generated based on similarity levels. They used the Pearson correlation similarity measure and varied it between 0.3, 0.5, 0.7, and 0.9, and the size between 3, 5, and 8. They generated 1 group for each evaluation condition (12 total). They reported that disagreement between group members impacted the quality and efficiency and could be exploited to increase the effectiveness of the group recommendations. In this evaluation, they did not use a baseline for evaluation, they just measured and compared performance between the various algorithms.

This section overviewed some of the related work that evaluated group recommender systems utilizing user studies and synthetic groups. In both evaluation approaches researchers had to make judgements in the experiment setup in order to measure system performance. In comparison to individual-based systems, group-based systems introduce extra overhead in the evaluation setup whether it is related to group formation, testing set creation, or establishing ground truth. In this paper we evaluate some of the choices that can be made in establishing the ground truth in evaluations that utilize synthesized groups.

3. EXPERIMENTAL SETUP

To establish ground truth for synthesized groups a need arises to model how the group consensus is achieved for any item given the individual group member's preference for that item. Previous work that needed to apply such models have mainly adopted the average strategy to best model the group's preference for any test item as a baseline for evaluation. In this evaluation, we investigate the tradeoffs in potential outcomes when groups employ different strategies. We aim to understand the impact of selecting different models as the *AGPM* in the evaluation of a group recommender system. We explore outcomes using some of the most commonly used modeling strategies. We make a comparison between Average, Least Misery, and Most Happiness as defined in section 3.3. These strategies were varied on both the recommendation side and the actual (ground truth) side. We also analyze their performance with respect to group size and inner group similarity.

3.1 Recommendation Technique

For this analysis we have selected a very straightforward and commonly used recommendation technique; traditional user-based, Collaborative Filtering [9, 22]. This is employed for individual user predictions, that are then aggregated for group recommendation. The basis for this approach is to calculate a neighborhood similarity between users a and b, w_{ab}, using Pearson correlation:

$$w_{ab} = \frac{\sum_{i=1}^{n}[(r_{ai} - \overline{r}_a)(r_{bi} - \overline{r}_b)]}{\sqrt{\sum_{i=1}^{n}(r_{ai} - \overline{r}_a)^2 \sum_{i=1}^{n}(r_{bi} - \overline{r}_b)^2}} \qquad (1)$$

To generate predictions, a subset of the nearest neighbors of the active user are chosen based on their correlation. We then calculate a weighted aggregate of their ratings to generate predictions for that user. We use the following formula to calculate the prediction of item i for user a:

$$p_{ai} = \overline{r}_a + \frac{\sum_{b=1}^{n}[(r_{bi} - \overline{r}_b) \cdot w_{ab}]}{\sum_{b=1}^{n} w_{ab}} \qquad (2)$$

Herlocker et al. [9] noted that setting a maximum for the neighborhood size less than 20 negatively affects the accuracy of the recommender systems. They recommend setting a maximum neighborhood size in the range of 20 to 60. We set the neighborhood size to 50.

3.2 Data Set and Group Generation

To evaluate the accuracy of an aggregated predicted rating for a group, we use the MovieLens dataset of 100,000 ratings from 943 users on 1682 movies. In creating synthetic groups for evaluation, we varied group size and degree of similarity among group members. The group sizes were varied from 2 to 5, and we defined three inner group similarity levels:

· high: 0.5 <= inner group similarity < 1
· medium: 0 <= inner group similarity > 0.5
· low similarity: -1 <= inner group similarity >= 0

The inner similarity correlation between any two users i, j, belonging to group G, is calculated using the Pearson Correlation as defined in equation 1.

We then created 5000 groups for each of the defined size (2,3,4,5) and similarity (high, medium, low) levels. To evaluate different models for the *AGPM*, we randomly selected a test item from the set of commonly rated items for each group. For each group, one item has been identified as the test item. Experimental data is collected for accuracy measurement of the prediction on those test items.

For prediction calculation and evaluation we used the leave-one-out approach where only the test item rating for that group's members is left out of the training set. For example, given a group of $\{user_1, user_2, user_3\}$ and test item IDs of {50, 100}; when calculating the prediction of item 50 for $user_1$, only the ratings of item 50 for users $\{user_1, user_2, user_3\}$ are taken out of the data set to form the training set for that group. Then predictions for that item are generated using this training set and so on for the other groups and test items. The computed, predicted ratings for each group member are then aggregated using the different models for group recommendations (*RGPM*) to produce a final group predicted rating.

3.3 Group Aggregation Strategies

Various group modeling strategies for making recommendations have been proposed and tested to aggregate the individual group user's preferences into a recommendation for the group. Masthoff [14] evaluated eleven strategies inspired by social choice theory. For clarity, we focus on three representative strategies: average strategy, least misery, and most happiness.

- Average Strategy: This is the basic group aggregation strategy that assumes equal influence among group members and calculates the average rating of the group members for any given item as the predicted rating.

Let n be the number of users in a group and r_{ij} be the rating of user j for item i, then the group rating for item i is computed as follows:

$$Gr_i = \frac{\sum_{j=1}^{n} r_{ij}}{n} \tag{3}$$

- Least Misery Strategy: This aggregation strategy is applicable in situations where the recommender system needs to avoid presenting an item that was highly disliked by any of the group members, i.e., that goal is to please the least happy member. The predicted rating is calculated as the lowest rating for any given item among group members and computed as follows:

$$Gr_i = \min_j r_{ij} \tag{4}$$

- Most Happiness: This aggregation strategy is the opposite of the least misery strategy. It applies in situations where the group is as happy as their happiest member and computed as follows:

$$Gr_i = \max_j r_{ij} \tag{5}$$

3.4 Accuracy Measurement

To evaluate the accuracy of a predicted rating computed for a group across different test conditions, we use the root-mean-square error (RMSE) [10]. RMSE measures the differences between values predicted by a model and the actual values. To do so, we compared the group predicted rating calculated for the test items, using the three aggregation methods as the *RGPM*, to a model of the actual rating across the different group sizes and inner-group similarity levels.

4. RESULTS

To analyze the impact of the choice of a model as the actual group preference on the accuracy levels of predictions generated for the group, we compare the accuracy levels of predictions generated using the three different models defined in section 3.3. Surely changing the baseline will change the accuracy results but we are interested in the size, extent, and trends of such differences across different group conditions.

4.1 Relationship between the AGPMs

Figures 1 to 3 show the RMSE values for the various conditions. To evaluate the difference in RMSE values we use the two-tailed, t-test for statistical significance with a p value of 0.01. For each group similarity level (high, medium, low) and *AGPM* (LM, MH, Average), we compared the *RGPMs* (LM, MH, Average) to each other. So, for each pair of *AGPM* (Avg/LM, Avg/MH, LM/MH), we compared the various *RGPMs* for each group similarity level and group size (2,3,4,5).

For this evaluation there were 324 conditions (3 *AGPM* pairs x 9 *RGPM* pairs x 4 group sizes x 3 similarity levels). Of those only 4 had a p value > 0.01. Examining those 4 non-significant relationships further, they are between the LM_AGPM and MH_AGPM. Thus, as expected, there are significant differences in most RGPM evaluations, depending on the active *AGPM* baseline. For the LM_AGPM (Figure 2) results indicate that the LM_RGPM performs best across all group inner-similarity levels and the different group sizes. The difference between the RGPMs was significant for each group inner-similarity level and each group size. The same results are found for the MH_AGPM (Figure 3) and the Avg_AGPM (Figure 1).

Indeed, we would be surprised, were it not the case, that evaluation outcomes were biased when the aggregation strategy used for the *RGPM* is also used for the AGPM. If we knew which aggregation strategy the group actually used, applying it on the recommendation side, one would expect that it would result in more accurate recommendations. From these results we can conclude that the choice of a baseline to represent the AGPM does indeed matter and the commonly adopted average model, as a baseline, does not necessarily provide an ideal baseline for evaluating group-based recommender systems.

4.2 Relationship between the RGPMs

In comparing evaluation outcomes, we examine the outcomes across individual conditions and the correlation across all group sizes. Tables 1-3 show the correlation among accuracy values between the three aggregation models when used as the *AGPM* as well as the *RGPM* for the three defined inner group similarity levels.

Table 1: Correlations between the RGPMs using the different AGPMs for groups with low similarity levels

		Avg_RGPM	LM_RGPM	MH_RGPM
LM – MH	Size 2	-0.1101	-0.0900	-0.1470
	Size 3	-0.0332	-0.0585	-0.0985
	Size 4	0.0216	-0.0403	-0.0557
	Size 5	0.1806	0.0783	-0.0040
LM – AVG	Size 2	0.4320	0.1751	0.7026
	Size 3	0.1738	-0.1626	0.6401
	Size 4	0.1854	-0.1817	0.6181
	Size 5	0.1030	-0.0827	0.5798
MH – AVG	Size 2	0.2761	0.5788	0.0398
	Size 3	0.2303	0.6300	-0.1284
	Size 4	0.1923	0.5853	-0.1401
	Size 5	0.1309	0.4684	-0.0716

Table 2: Correlations between the RGPMs using the different AGPMs for groups with medium similarity levels

		Avg_RGPM	LM_RGPM	MH_RGPM
LM – MH	Size 2	-0.1823	-0.1356	-0.1420
	Size 3	-0.1428	-0.0907	-0.1054
	Size 4	-0.0923	-0.0624	-0.1309
	Size 5	-0.0202	-0.0444	-0.0630
LM – AVG	Size 2	0.5158	0.2703	0.7402
	Size 3	0.2737	0.0087	0.6458
	Size 4	0.1291	-0.0366	0.6374
	Size 5	0.0452	-0.0486	0.5754
MH – AVG	Size 2	0.3134	0.6242	0.1733
	Size 3	0.2730	0.6483	0.0904
	Size 4	0.2471	0.6101	-0.0972
	Size 5	0.1963	0.5836	-0.0536

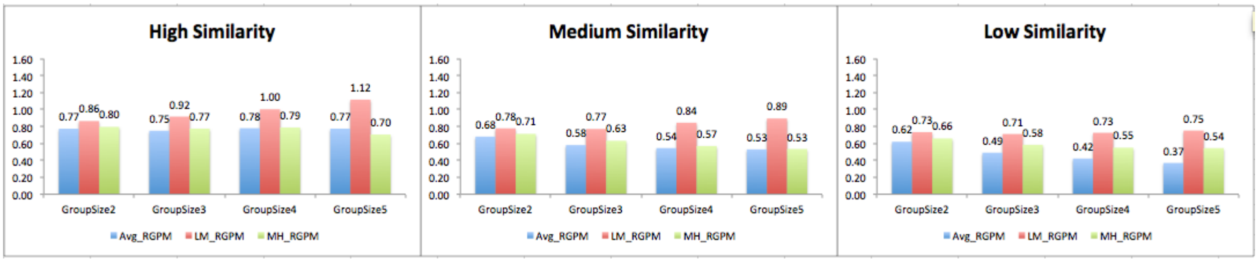

Figure 1: RMSE for Avg_AGPM

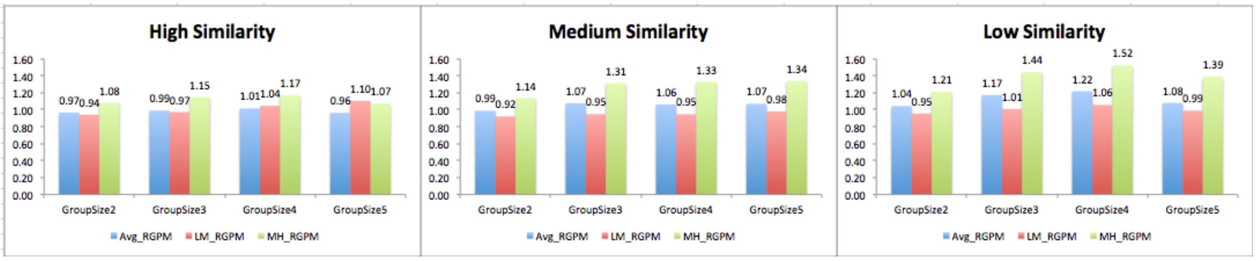

Figure 2: RMSE for LM_AGPM

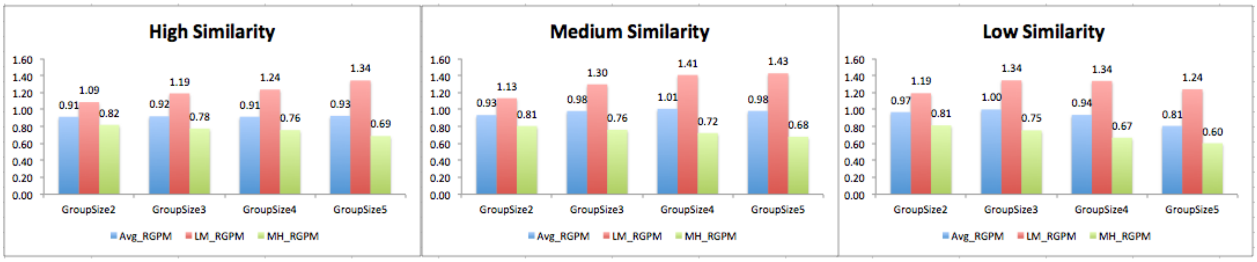

Figure 3: RMSE for MH_AGPM

Table 3: Correlations between the RGPMs using the different AGPMs for groups with high similarity levels

		Avg_RGPM	LM_RGPM	MH_RGPM
LM – MH	Size 2	-0.1233	-0.1216	0.0656
	Size 3	-0.2017	-0.0872	0.0535
	Size 4	-0.1425	-0.0657	0.2397
	Size 5	-0.1368	-0.0894	0.2451
LM – AVG	Size 2	0.7159	0.5238	0.8418
	Size 3	0.6006	0.3620	0.8011
	Size 4	0.5751	0.3878	0.7814
	Size 5	0.5544	0.3966	0.7771
MH – AVG	Size 2	0.4186	0.6336	0.4520
	Size 3	0.3168	0.6923	0.3800
	Size 4	0.3487	0.6528	0.6083
	Size 5	0.3007	0.6035	0.5794

Examining the correlations between the LM_AGPM and the MH_AGPM, we can see that these two approaches are mostly negatively correlated. The correlation is strongest for the MH_RGPM for groups with a high inner-group similarity level. One explanation for that is that, as the group members are highly similar, they tend to give items ratings that are similar or close. In that case, both aggregation models used as a baseline might yield comparable evaluation results.

When comparing the LM_AGPM to the Avg_AGPM we can also see a similar trend. For groups that have highly similar members the evaluation outcomes, using these two *AGPMs* as a baseline, are highly correlated for all group sizes and *RGPMs*. Since the group members are similar they tend to rate similarly or very close to the group's average rating for that item.

As the group sizes increase, for groups with medium and low similarity levels, the Avg_RGPM and LM_RGPM can result in different evaluation outcomes. We can see that all the lowest correlations are for the LM_RGPM and appear in groups of medium and low similarity. This can be because, as the group's inner similarity level decreases, the chances that they rate an item differently increases which makes it further from the average rating of the group member's for that item. Another factor is, as we pointed out earlier, the LM_RGPM favors the lowest rating and that might be further away from the average group rating for that same item. This also highlights the bias effect introduced when comparing results with the same aggregation strategies used on the recommendation and evaluation sides. Here we are comparing the LM_RGPM using the LM_AGPM as a baseline.

We attribute this relationship to the fact that these two aggregation strategies are on opposite sides. The LM_AGPM favors the lowest rating while the MH_AGPM favors the highest rating. We can see the conditions where the RGPMs are negatively correlated between the LM_AGPM and the Avg_AGPM. We can see that all the negative correlations are for the LM_RGPM and appear in groups of medium and high similarity. This can be because, as the groups inner similarity level increases, there is a greater chance that they rate an item more similarly, which makes it closer to the average rating of the group member's for that item.

Another factor is, as we pointed out earlier, the LM_RGPM favors the lowest rating and that might be further away from the average group rating for that same item. This also highlights the bias effect introduced when comparing results with the same aggregation strategies used on the recommendation and evaluation sides. Here we are comparing the LM_RGPM using the LM_AGPM as a baseline. We notice the same trends when comparing either MH or LM to the Avg AGPM. We see the low correlations more in the low similarity groups for the MH_RGPM. This might indicate that, for low similarity groups, using the MH_RGPM might not result in satisfactory recommendations to all the group members. Here the bias effect is also highlighted since the MH_AGPM is one of the baselines used here.

5. CONCLUSION

In this paper we analyzed different choices of an aggregation strategy to model the actual group preference when evaluating the accuracy performance of a group-based recommender system using synthesized groups. We compared the results of using the Avg_AGPM, LM_AGPM and MH_AGPM as a baseline to evaluate the performance of three representative aggregation strategies as the RGPMs using synthesized groups with various degrees of inner similarity and size. Results show that the choice of an AGPM, in this context, results in different evaluation outcomes. The choice of an AGPM can also introduce a bias, particularly when the same aggregation strategy is used on the recommendation side.

In this paper we aimed to show that modeling the actual group preference does matter when evaluating group-based recommenders, and different models provide different results. Developers of such systems need to consider the tradeoffs when choosing a baseline for evaluation. Here we demonstrated the differences between some of these choices. Overall, this work has helped to extend the coverage of group recommender evaluation analysis, and we expect this will provide a novel point of comparison for further developments in this area.

6. REFERENCES

[1] S. Amer-yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *Proceedings of The Vldb Endowment*, 2, 2009.

[2] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence*, 2003.

[3] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[4] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 111–118, New York, NY, USA, 2010. ACM.

[5] Y.-L. Chen, L.-C. Cheng, and C.-N. Chuang. A group recommendation system with consideration of interactions among group members. *Expert Syst. Appl.*, 34, 2008.

[6] I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman. A group recommender system for tourist activities. In *Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, 2009.

[7] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM International Conference on Supporting Group Work*, 2010.

[8] D. Goren-Bar and O. Glinansky. Fit-recommending tv programs to family members. *Computers & Graphics*, 28(2):149 – 156, 2004.

[9] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5, 2002.

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.

[11] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, 2004.

[12] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The adaptive web*. 2007.

[13] J. Masthoff. Group modeling selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14, 2004.

[14] J. Masthoff. Group recommender systems: Combining individual models. In *Recommender Systems Handbook*. 2011.

[15] J. F. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. pages 1–10, 2002.

[16] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Cats: A synchronous approach to collaborative group recommendation. 2006.

[17] N. A. Najjar and D. C. Wilson. Evaluating group recommendation strategies in memory-based collaborative filtering. In *Proceedings of the ACM Recommender Systems Conference Workshop on Human Decision Making in Recommender Systems*, RecSys '11. ACM, 2011.

[18] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: a recommender system for groups of users. In *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*,

2001.

[19] L. Quijano-SÃąnchez, D. Bridge, B. Diaz-Agudo, and J. Recio-Garcia. Case-based aggregation of preferences for group recommenders. In B. Agudo and I. Watson, editors, *Case-Based Reasoning Research and Development*, volume 7466 of *Lecture Notes in Computer Science*, pages 327–341. Springer Berlin Heidelberg, 2012.

[20] L. Quijano-SÃąnchez, D. Bridge, B. Diaz-Agudo, and J. Recio-Garcia. A case-based solution to the cold-start problem in group recommenders. In B. Agudo and I. Watson, editors, *Case-Based Reasoning Research and Development*, volume 7466 of *Lecture Notes in Computer Science*, pages 342–356. Springer Berlin Heidelberg, 2012.

[21] J. A. Recio-Garcia, G. Jimenez-Diaz, A. A. Sanchez-Ruiz, and B. Diaz-Agudo. Personality aware recommendations to groups. In *Proceedings of the third ACM conference on Recommender systems*, 2009.

[22] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *1994 ACM Conference on Computer Supported Collaborative Work Conference*, 1994.

[23] M. Salamó, K. McCarthy, and B. Smyth. Generating recommendations for consensus negotiation in group personalization services. *Personal and Ubiquitous Computing*.

[24] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*. 2010.

Evaluating Conversational Recommender Systems based on Preference Based Feedback

Saurabh Gupta
Department of Computer Science & Engineering
IIT Madras, Chennai, India
sgupta@cse.iitm.ac.in

Sutanu Chakraborti
Department of Computer Science & Engineering
IIT Madras, Chennai, India
sutanuc@cse.iitm.ac.in

ABSTRACT

Conversational recommender systems are a special kind of knowledge based systems which iteratively build a conversation with a user, showing her a set of products at each iteration and seeking her feedback on those items. Feedback given by the user is then used to generate a new set of recommendations and this process continues until the user settles on a product. Preference based feedback (PBF) is one way to get user feedback, where the user selects one item over the others at each interaction with the system. Traditionally, while evaluating such systems, a target product is fixed for a query, based on *weighted similarity model* of utility, and then an agent is simulated, such that at each step, the product which is maximally similar to the target is chosen as the next preference of the simulated agent. The next set of recommendations is generated based on the utility function being evaluated. Efficiency is considered to be of utmost importance (algorithms reaching the target in very few interaction cycles are preferred) in these situations. But the quality of the actual journey leading to the target is seldom considered as a metric to evaluate against. To simulate a realistic scenario where the user does not behave optimally(always selecting most similar product to the target), noise is added arbitrarily at each step. We propose evaluation metrics for PBF based conversational recommenders which also take into account the quality of recommendations served to the user en route the target. Also, we propose an evaluation methodology, which introduces noise into the interaction process in a principled manner as opposed to an arbitrary manner, as is done in the traditional evaluation methodology. Moreover, we show that evaluation methodology proposed in this paper is agnostic to the utility function used.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance Evaluation*

General Terms

Measurement, Performance, Standardization

BARS'13, August 1, 2013, Dublin, Ireland.
.

Keywords

Knowledge based Recommendation,Preference based Feedback, Case Based Recommendation, Evaluation Methodology, Evaluation Metrics

1. INTRODUCTION

One of the most popular approaches of generating recommendations is Collaborative Filtering [2][5], which is a major component in many commercial recommendation systems. However, there are a variety of scenarios where Collaborative Filtering might not be the best approach to generate effective recommendations. For example, as pointed by Burke[1], in cases where we want to purchase houses, cars or computers, Collaborative Filtering might be inadequate because of the sparsity of ratings in such domains. This is because we do not purchase these commodities very often. In these kinds of domains, the user would typically want to specify some of her information needs explicitly. This kind of preference expression is not particularly amenable to the usual way in which Collaborative Filtering approaches solicit an input from a user(they usually want ratings from the user).

These issues can be addressed to some extent by Knowledge Based recommender systems. Case based Recommender systems are a special kind of knowledge based systems which make use of similarity measures (between user specification and a particular item or between two items) to recommend items to a user[6]. Let P be the user query or a product currently selected by the user, then the utility of another product Q is given as

$$sim(P,Q) = \sum_i w_i \times localsim(P_i, Q_i) \tag{1}$$

where $localsim(P_i, Q_i)$ is the local similarity between the i^{th} attributes of P and Q and w_i refers to the weight assigned to the i^{th} attribute. We call this model of utility computation of items as the *weighted similarity model*. Case based Recommender systems can be further divided into *Single Shot Systems* and *Conversational systems*. While single shot retrieval systems retrieve products similar to the user query in one go, recommendation in conversational systems takes place in an adaptive and interactive manner spread across multiple iterations. At each interaction step with the user, they solicit some kind of feedback from the user which is then used to generate recommendations for the next cycle. Single shot systems assume that the user has her preferences clearly defined in her head, which may not be always true. In situations when the user only has a vague idea about her preferences, conversational systems provide a way to explore the product space effectively helping the user form her preferences in the process.

Preference based feedback (PBF)[8] is a particular feedback elicitation method used in conversational systems, where the user has to indicate a preference for one product over the others (choose one among the k products shown) at each interaction with the system. PBF is an attractive mode for getting user feedback as it exerts a limited cognitive load on a user during her interaction with the system. Cognitive load can be understood as the amount of effort exerted by the user to give her feedback. Other modes of feedback like *asking direct questions from the user* or *critiquing* (user constrains a particular feature of a recommended product) demand much more on the part of the user mentally.

In this paper, we point out some of the drawbacks associated with the traditional metrics and the methodology employed to evaluate conversational recommender systems based on PBF. Note that in this paper, we concern ourselves with the off-line evaluation(not relying on user studies) that is done to evaluate PBF based recommenders. In the process, we propose new metrics and methodology which can be used to evaluate PBF based conversational systems. Section 2 discusses the background and related work on how evaluation is traditionally carried out in a typical PBF based conversational recommender system (CRS) and sets the stage for the proposed metrics and methodology discussed in Section 3.

2. BACKGROUND AND RELATED WORK

Traditionally, CRSs based on PBF are evaluated using a "simulated" user [10, 4, 9]. Each CRS has an underlying utility function which governs the items that are shown to the user at each interaction. For all purposes in this paper, we assume the utility function to be the *weighted similarity model* described in Eq. 1. Most PBF based CRSs follow a leave-one-out methodology, where each product is first removed from the products database and used to generate queries of various lengths(comprising one or many features of the left out product). This leads to the generation of both underspecified and over specified queries. For a query generated in this way, the target product is that item which is most similar(corresponding to *weighted similarity model*) to the product from which the corresponding query was generated. During each interaction cycle, the simulated user chooses that product from among the k shown products, which is most similar to the target product. In the next cycle, the system generates another set of k products based on its underlying utility function(same as Eq. 1) and the "simulated" user again chooses the most similar product to the target and this process goes on till the target product gets returned in one of the interaction cycles. Such an evaluation methodology is not just limited to CRSs using PBF. Some recommendation approaches based on critiquing as discussed in [3, 7] also use a similar evaluation methodology.

Recommendation efficiency is usually measured in terms of the average number of interaction cycles taken to reach the target as well as the average number of unique items shown en route the target product [10, 4, 9, 3, 7]. Algorithms which take lesser average cycles and show lower number of unique items on an average are considered to be attractive to the user from a cognitive load viewpoint. One problem with such a consideration of lower average cycles and average items is that it does not conclusively prove the superior performance of one recommendation approach over the other. This is because a particular approach might have lower average numbers due to small numbers incurred on some of the queries[4], but inferior performance across many other queries. The authors in [4] proposed a metric called the number of "wins" that one competing algorithm achieves as compared to the other. Specifically, the number of queries for which algorithm A1 takes lesser(or equal) number of cycles than A2, is the number of wins achieved by A1 against A2.

The"simulated" agent always selecting the most similar product to the target at each interaction cycle implies that the agent is behaving optimally, always selecting the best preference item at every step. To relax this assumption, preference noise was added at every interaction cycle in [9], thus simulating an agent which can issue sub-optimal preferences. For example, noise of 5% in an interaction cycle means that the similarities of each of the k products in the current cycle, to the target, are perturbed by +/-5%, which might end up changing the ordering of the items shown and might result in the selection of an item which might not have been the most similar item to the target.

The traditional evaluation methodology for PBF based CRSs has the following drawbacks:

- The assumption that the "simulated" agent selects the maximally similar product to the target at each interaction is unrealistic.

- We observe that the noise addition procedure discussed above is arbitrary. This is because the similarities of the products shown in a particular cycle, to the target, are perturbed randomly. There is no principled way in which the similarities get changed for the non-optimal option to get selected.

Also, most of the evaluation metrics for CRSs based on PBF are variants to estimate the cognitive load on the user. There has been not much work done to characterize the quality of recommendation during each cycle in the context of CRSs. But we believe that the journey that the user undergoes in order to reach the target is also important when we seek to evaluate conversational recommender systems. The need to evaluate quality of recommendations per cycle can be motivated through the following example: Suppose, during the first few interactions with the system, the user is shown items in which she is not interested, there is high chance that she might quit the interaction with the system, even though the interaction might have led to the target if she would have continued the interaction for another couple of iterations.

3. PROPOSED METHODOLOGY AND METRICS

There is need to simulate artificial users who do not always behave optimally . Note that the word "optimal" here refers to the ability of the selected product to lead to the target product. Traditionally, in the domain of case based recommender systems evaluation, the target corresponding to a query is estimated based on the *weighted similarity model* described earlier. Hence this ability of a choice to lead to the target product is usually measured in terms of its similarity to the target. So among a set of options shown to the user, the "optimal" choice is supposed to be the one which is most similar to the target. The "simulated" agent selecting the optimal choice at every interaction (the evaluation methodology generally followed) is a kind of behaviour which might not always be observed in real life settings. Various biases exist in human decision making which might not always be amenable to optimal decision making. We propose a more principled way to simulate a user interaction where the user can express non-optimal preferences. In this section, we also propose some metrics for quality evaluation of individual cycles of items during conversational recommendation.

3.1 Tree based simulation of non-optimal user behaviour

We propose a tree based probabilistic methodology to evaluate conversational recommender systems based on PBF. We explain our proposed methodology for evaluation with respect to Fig. 1 in which "a" represents the query that is fired by the "simulated"

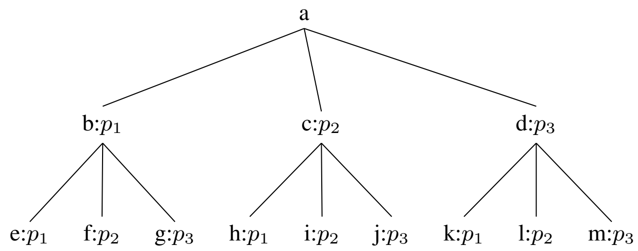

Figure 1: Evaluation framework based on probabilities

user to start off an interaction with the system. Any parent node serves as the query for the algorithm being evaluated and its children represent those products which are returned by the algorithm in response to that query, based on its utility function, with the product with highest utility being the leftmost child and the utility of children decreases as we move from left to right. As stated earlier, although we will be using *weighted similarity* as the utility function for demonstration purposes in our paper, the proposed methodology is agnostic to the type of utility function used. Therefore, "b", "c", "d" are the products most similar to query "a" and "h", "i", "j" are the three most similar products to product "c", with utility(similarity) of "h" higher than that of "i", which in turn is higher than that of "j". The number of children of each node is equal to the number of products shown at each iteration by the system to the user. Hence, the interaction modelled by Fig. 1 is one in which the user is shown three products at each interaction. p_1, p_2 and p_3 are the probabilities with which the product with whom they are associated would be selected by the "simulated" agent in the current iteration. In this setting, we fix $p_1 > p_2 > p_3$. This is because p_1 is always associated with a product which has the highest utility followed by the products associated with p_2 and p_3. The intuition behind this setting is as follows: Any given utility function which we evaluate in this methodology will rank products during the expansion of each node(product) in the tree. Since the leftmost node in the expansion of a parent node has the highest utility, assuming that the user's model of utility is similar to utility model of the algorithm being evaluated, the probability of a user selecting that product is higher as opposed to the other products(the middle and the rightmost children of any parent node) shown along with it but ranked below it by the utility function. A particular path, $\{a_1, a_2, ...a_n\}$ in the tree corresponds to a user trail starting from query or product a_1 and ending at product a_n. For example - the path {a,b,g} in Fig. 1 represents the path taken by the "simulated" user to reach "g" if she starts from her query "a". As explained in Section 2, each starting query is associated with a target product that has to be reached. Referring to Fig. 2, the starting query is "a" and let us assume that "s" is the target for this query, chosen as explained in Section 2. As we can see from Fig. 2, one of the paths to the target from the query is {a,b:p_1,f:p_2,s}. We can associate a semantic meaning to this path - the "simulated" user starts with the query "a", then selects product "b" with probability p_1, after which she selects product "f" with probability p_2 in response to which the system shows target product "s" seeing which the user ends the interaction. We can quantify the probability of a user landing up at the target product "s" via the above path as :

$$Pr(\{a, b : p_1, f : p_2, s\}) = p_1 \times p_2 \tag{2}$$

where p_1 is the probability of selecting "b" after query "a" and p_2 is the probability of selecting "f" after the systems returns a set of three products which are most similar (have highest utility) to "b", which was selected earlier in the interaction. Now there can be multiple paths from the query node which can lead to the target product. Therefore the performance metric, M, of a particular algorithm in our methodology can be aggregated as

$$M = \sum_{i \in paths} Pr(i) \tag{3}$$

where $paths$ is the set of all paths from the query node to the target node and $Pr(i)$ is computed as described in Eq. 2. The more the value of M, the better the algorithm in its ability to the lead to the target product. As we can see from the above description, our evaluation methodology involves a principled way of assigning probabilities to individual children of a query product, as opposed to the random noise introduction done in the traditional methodology. The principle followed is - if the "simulated" agent behaves according to the utility function of the algorithm being evaluated in our methodology, then there is a high chance that she would select the leftmost child, with the probability of a child product getting selected decreasing as we go from left to right ($p_1 > p_2 > p_3$). Another difference of our methodology(G) with the traditional one (T) is that in T, the selection of a product at each interaction step is governed by the *weighted similarity model*. As a result, during each interaction, a single product gets selected and there is a single path from the query to the target. On the other hand, in G, there are probabilities associated with each product shown in an interaction cycle, and the "simulated" agent selects all these products according to those probabilities. As a result, there are multiple paths from the initial query to the target in G. G is not coupled with any type of utility function during the process of selecting a product at each interaction.

Note that in our proposed methodology, the number of paths from the query to the target node can potentially be very high. A product that appears in one branch can appear in another branch too. For example - in Fig. 1, product "b" can occur in the subtrees rooted at "c" as well as "d". Also note that we drew the tree up to only three levels for illustration purposes; ideally we can keep expanding all the nodes till all paths to the target have been computed. But the number of paths is combinatorially huge. We limit this complexity with the help of the following two observations:

- **a product node cannot appear in any of the subtrees rooted at its direct children**. For example - in Fig. 1, "b" can never occur in subtrees rooted at "e", "f" and "g". This is a perfectly reasonable assumption in the context of conversational recommendation because once a product has been selected by a user at a particular iteration, it need not be shown again to him in any of the subsequent iterations. Note that "b" can occur in subtrees rooted at children of "c" and "d".
- **expansion of nodes is stopped after certain of number of levels k, with k not being very small**. For example we can fix that the number of levels that we will expand up to will be 12. This is reasonable because after a point, the product of probabilities that is calculated according to Eq. 2 will become negligible and would not make any significant contribution to the sum of all path probabilities calculated according to Eq. 3.

3.2 Quality of Recommendations per cycle

As indicated earlier, most of the conversational systems for recommendation are evaluated according to the cognitive load experienced by the user, in terms of the number of cycles taken to reach the target. But the quality of the journey undertaken is usually ignored. The ideas presented in this subsection are suggestive and

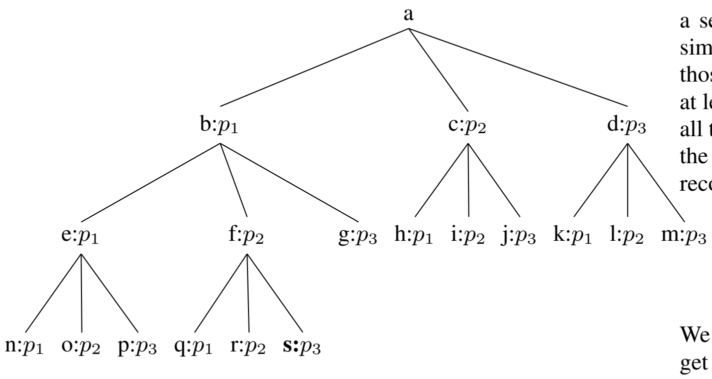

Figure 2: Reaching the target product “s" marked in bold

not prescriptive and need to be tailored to the application domain if the need arises. We propose the following three metrics to take into account the quality of the journey as well.

Similarity to the target:

The average similarity of all the items shown in a particular iteration, to the target, is an important indicator of how close the user is to the target. Let the set of k products shown to the user be denoted by X and let the target product be denoted by t. The quality of recommendations per cycle, Q, can then be quantified as :

$$Q = \frac{\sum_{p \in X} sim(p,t)}{k} \tag{4}$$

Q aggregated over all the cycles gives a measure of quality of recommendation across the entire journey of the user to reach the target.

Combination of similarity and diversity:

Although the importance of diversity as part of the utility function has been well documented[8][9], when it comes to evaluating a conversational recommender system, only average number of cycles and unique items are reported. We combine the notion of average similarity of items in a cycle to the target with the average dissimilarity between them. Let us consider $X = \{p_1, p_2 ....p_k\}$, containing k products that are shown to the user in a particular iteration and let t be the target product.

$$sim = \frac{\sum_{u \in X} sim(u,t)}{k} \tag{5}$$

$$div = \frac{\sum_{i \in \{1,2,3...k\}} \sum_{j \in \{1,2,3...k\}} (1.0 - sim(p_i, p_j))}{\frac{k}{2} * (k-1)} \tag{6}$$

where $(1.0 - sim(p_i, p_j))$ measures the dissimilarity between products p_i and p_j. The metric combining both similarity and diversity, SD, is given by:

$$SD = sim \times div \tag{7}$$

The metric, SD, when aggregated over all the cycles quantifies the quality of recommendation during the journey undertaken by the user to reach the target.

Number of items within a similarity threshold:

In [9], the authors suggested that users usually might get satisfied by any one of a group of items which are sufficiently similar to the target product. But again, they used it to build a bigger set of possible target items and terminated the user interaction when any one of these items was shown to the user. Finally, the average number of cycles were reported. We use this notion of multiple targets to develop a metric concerned with quality. To generate a set of multiple targets a similarity threshold is kept. An 80% similarity threshold means that the list of target products include all those products whose similarity with the original target product is at least 0.8. Let $T = \{t_1, t_2, t_3 .....t_n\}$ represent the set containing all the target products and let C represent the set of items shown to the user currently. We can then define a measure Q for the current recommendation cycle as

$$Q = \frac{|C \cap T|}{|C|} \tag{8}$$

We sum up Q values across all the iterations till the original target is reached. More the value of Q, better is the algorithm.

4. CONCLUSIONS

In this paper, we proposed an evaluation methodology for PBF based conversational recommender systems, which introduces noise into the product selection process in a principled way. Additionally, the proposed evaluation methodology is agnostic to a particular utility function. We also suggested the use of quality metrics for individual cycles in conversational systems, instead of just concentrating on efficiency measures.

5. REFERENCES

[1] R. Burke. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*, page 2000. Marcel Dekker, 2000.

[2] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, Dec. 1992.

[3] M. S. Llorente and S. E. Guerrero. Increasing retrieval quality in conversational recommenders. *IEEE Trans. Knowl. Data Eng.*, 24(10):1876–1888, 2012.

[4] L. McGinty and B. Smyth. Comparison-based recommendation. In *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, ECCBR ’02, pages 575–589, London, UK, UK, 2002. Springer-Verlag.

[5] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW ’94, pages 175–186, New York, NY, USA, 1994. ACM.

[6] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35. 2011.

[7] M. Salamó, J. Reilly, L. McGinty, and B. Smyth. Knowledge discovery from user preferences in conversational recommendation. In *Proceedings of the 9th European conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD’05, pages 228–239, Berlin, Heidelberg, 2005. Springer-Verlag.

[8] B. Smyth. Case-Based Recommendation. In *The Adaptive Web*, pages 342–376. Springer, 2007.

[9] B. Smyth and L. Mcginty. The power of suggestion. In *In IJCAI*, pages 127–132. Morgan Kauffman, 2003.

[10] J. Zhang and P. Pu. Refining preference-based search results through bayesian filtering. In *Proceedings of the 12th international conference on Intelligent user interfaces*, IUI ’07, pages 294–297, New York, NY, USA, 2007. ACM.