Towards Increasing the Coverage of Interactive Recommendations

Behnam Rahdari¹, Peter Brusilovsky¹, Branislav Kveton²

¹University of Pittsburgh, Pittsburgh, PA
²Amazon
behnam.r@pitt.edu, peterb@pitt.edu, bkveton@amazon.com

Abstract

An interactive recommender system pursues two somewhat contradictory goals. On one hand, the system should provide highly relevant recommendations with the best match to the overall user needs. On the other hand, the recommendations should be sufficiently diverse to cover a range of users’ possible interests. Such recommendations increase chances that the user finds items that match their context while also informing the system which items are currently most important. In this paper, we present a ranking approach that balances the demands of relevance and coverage. We evaluate the approach on two problems, of advisor and movie recommendations, where the immediate needs of the user are likely to be diverse. Our approach considerably increases chances that the user finds relevant items in the first few steps of the recommendation dialog.

Keywords: Recommender Systems; Diversity

Introduction

The majority of research on recommender systems focuses on developing algorithms for “perfect” item ranking bringing items that are considered most relevant on the top of the recommendation lists (Cremonesi, Koren, and Turrin 2010). This ranking is generated based on past user ratings and actions that are considered as ultimate knowledge about the user. However, the data about user interests is never complete and precise, and their true interests are not always easy to predict reliably. Moreover, the majority of users of modern recommender systems have multiple interests. They can watch movies of different genres, explore restaurants of different cuisines, and read research papers on a range of research topics. Depending on the situation, goals, and immediate needs, these users might prefer items reflecting different aspects of their interests, for example pick up a genre of a movie to watch based on their mood (Adomavicius et al. 2011). As a result, the “best overall” ranked list generated by traditional recommender systems will not bring truly relevant results for these users in many cases.

It has been recognized that the way to address this problem is human-AI collaboration where a recommender system and a user work together to obtain a list of items that is of interested to the user in the current situation (Chen and Pu 2012; He, Parra, and Verbert 2016). In this collaborative scenario, the recommendation process should not start with the “best overall” ranked list adapted to the dominated interests, but with a diversified list that offers a balanced coverage of various topics of interests. A coverage-focused list will enable the user to easily find candidates that are relevant to the current interests while informing the system what type of items they are looking for now. In this paper, we present a “greedy” approach to generate such a balanced list that considers both the strength of preferences and diversity.

We evaluate the approach on two problems, of advisor and movie recommendations, where the immediate needs of the user are likely to be diverse.

Related Work

A common approach to present recommendation to the users is known as Top-K ranking. With this approach, the user receives a single ranked list of recommendations where the items in the list are sorted based on the relevance to the user profile of interest. Top-K ranking has been studied by numerous researchers in the domain of recommender systems (Cremonesi, Koren, and Turrin 2010). However, as explained earlier, this approach is not suitable when the user profile consists of multiple topics of interest and when immediate interests of the users are not clear.

The problems of a single ranked list tuned to the “best overall” interests have been long recognized in research on recommender systems. The first stream of work that attempted to address this problem focused on combining relevance and diversity in item ranking. This work started in the field of information retrieval (Carbonell and Goldstein 1998) and was popularized in recommender systems (Ziegler et al. 2005). The diversity stream was motivated by an observations that stuffing the top of the recommended list with very similar items is not productive even though all these items have top relevance score. Numerous follow-up works explored a range of approaches to diversify the ranked list (Jambor and Wang 2010; Vargas and Castells 2011). Yet it was not specifically aimed to produce a balanced coverage of key interest topics and to increase user chances to provide useful feedback on the first step of human-AI dialogue. A more recent approach known as context-aware recommendation acknowledged that user’s current preferences are defined not just by the overall interest modeled by traditional
At each step and enabling the users to contribute more information process more efficiently by recommending a list of items. This feedback allows the system to better understand the user’s current preferences and gradually improve recommendations.

More recent research generalized this critique-based approach and attempted to make it more efficient using graphical user interfaces (GUI) in place of the natural language dialogue (Chen and Pu 2012). The use of GUI made the interaction process more efficient by recommending a list of items at each step and enabling the users to contribute more information through a more complex compound critiques (Smyth et al. 2004). Modern interactive recommender systems (He, Parra, and Verbert 2016) enhanced the use of GUI making it a centerpiece of human-AI interaction. Instead of a turn-taking dialog with a recommender agent, these systems offered users an opportunity to tune the results of recommendation continuously using sliders and other forms of direct manipulation (Bostandjiev, O’Donovan, and Höllerer 2012; Parra and Brusilovsky 2015). Yet focusing on empowering the user to refine recommendations through a dialog, interactive recommender systems rarely pay attention to making the first step of this dialogue (i.e., the starting item or list) sufficiently optimal for the user.

**Settings**

We investigate the problem of coverage in interactive recommendation in a typical modern context where items could be associated with multiple “interests” and users could favor several of these interests in parallel, although probably to a different extent and at different time. Depending on the domain, these interests could have different semantic nature. For example, it could be movie genres (such as action movies) or research topics (such as context-aware recommendation). We refer to these interests as topics.

To formalize our problem, we need to introduce notation. The set of all items is I and the set of all potential topics of interest is J. We index items by i ∈ I and topics by j ∈ J. For each item-topic pair, we define item-topic relevance s(i, j) ≥ 0, which represents the relevance of item i to topic j. To represent interests (preferences) of an individual user, we associate each topic j with weight w_j ≥ 0, which represents the preference of the user for topic j. Taken together, the set of these topic weights is a model of user interests (or simply user model).

The most natural approach to integrating Top-K ranking with multiple topics of interest is to rank items in descending order of \( \sum_{j \in J} w_j s(i, j) \), the item-topic relevances of an item i weighted the preferences for those topics. Formally, the total relevance for items I ⊆ I is

\[
  f_{\text{top}}(I) = \sum_{i \in I} \sum_{j \in J} w_j s(i, j),
\]

and the optimal set of items is

\[
  I_* = \arg\max_{I \subseteq \mathcal{I} : |I| = K} f_{\text{top}}(I).
\]

Although Top-K ranking is common in recommender systems, it is problematic in our setting, because it produces recommendations that overwhelmingly favor one topic over others, even if their weights are arbitrarily close to each other. To see this, consider the following example.

We have a user with two topics of interest, where w_1 = 1 and w_2 = 1 - ε for some small ε > 0. Therefore, the user prefers topic 1 only slightly more than topic 2. The pool of items is I = [10], where \( |n| = \{1, \ldots, n\} \) denotes the set of all integers from 1 to n. The first 5 items favor topic 1, \( s(i, j) = (0, 5, 0) \) for i ∈ [5]. The remaining 5 items favor topic 2, \( s(i, j) = (0, 0, 5) \) for i ∈ [10] \( \setminus [5] \). If we use Top-K ranking with K = 4, the optimal list would contain 4 items with relevances (0, 5, 0), because this yields the maximum objective value for (1) under the constraint of K = 4. This is clearly not a good recommendation. Roughly speaking, we would expect a recommendation with only slightly more items from topic 1, since w_1 = 1 and w_2 ≈ 1.

As mentioned earlier, the field of recommender systems suggested several solution for the observed overpopulation of the ranked list with dominant interests. Context-aware recommender systems would attempt to generate a better ranked list based on the current context, for example, observing that a user prefers to watch action moves on the weekdays and dramas at the weekend (Adomavicius et al. 2011). Conversational recommender systems might start with an “average best” item and engage user into a critiquing dialogue to determine what the users wants now (Burke, Hammond, and Young 1997). Interactive recommender system might allow the user to specify their current topic preferences (i.e., w_j) through a set of sliders (Parra and Brusilovsky 2015). Our approach introduced in the next section attempts to combine these ideas. On one hand, it assumes that a recommendation process is a human-AI collaboration that happens through interaction. On the other hand, rather than start with a single item of a sub-optimal ranked list in a hope to get to better recommendation through interaction, we want to produce an optimally balanced list that increases user chances to find relevant items and inform the recommender system about current preferences.

**Incorporating Diversity**

Mathematically, the problem with Top-K ranking is that the maximized objective function is modular in items. To address the problem, we borrow ideas from submodularity (Edmonds 1970; Nemhauser, Wolsey, and Fisher 1978). One objective function motivated by submodularity is

\[
  f_{\text{max}}(I) = \sum_{j \in J} w_j \max_{i \in I} s(i, j),
\]
If we used \( f_{\text{max}} \) in place of \( f_{\text{cov}} \) in the previous example, we are guaranteed to include at least one item with a very high similarity to at least one topic with a large weight. Specifically, for \( K = 4 \), we would have at least one item with relevance \((0.5, 0)\) and another with relevance \((0, 0.5)\). This yields the maximum value for (2). Unfortunately, the remaining 2 items could be arbitrary, as none of them changes the objective value.

**Proposed Solution**

We propose a new objective function,

\[
    f_{\text{cov}}(I) = \sum_{j \in \mathcal{J}} w_j \left[ \prod_{i \in I} (1 - s(i,j)) \right],
\]

where each item contributes to the objective. This addressed the main issue of (2), that not all items affect the objective. We justify this objective function from a probabilistic point of view in the next subsection.

It is easy to see that \( f_{\text{max}} \) and \( f_{\text{cov}} \) are monotone and submodular. Therefore, they can be maximized near-optimally by a greedy algorithm, which selects items with the highest modular. Therefore, they can be maximized near-optimally where each item contributes to the objective. This addressed the main issue of (2), that not all items affect the objective. We justify this objective function from a probabilistic point of view in the next subsection.

The guarantee on this maximization is that

\[
    f(I_K) \geq (1 - 1/e) \arg \max I \subseteq \mathcal{I} | |I|=K f(I)
\]

That is, the value of the greedily chosen set \( I_K \) of size \( K \) is at least 0.63 of that of the optimal set, which is a solution an to NP-hard combinatorial optimization problem (Nemhauser, Wolsey, and Fisher 1978).

We note that our objective (3) was proposed before in (El-Arini et al. 2009) to diversify blogs, where an item is a blog. We apply (3) to recommender systems and justify it using a click model. We also compare it to a simpler submodular objective (2), and systematically justify it through examples and experiments.

**Interpreting Objective**

Now we justify the objective function in (3). Specifically, we show that it is an expectation of the number clicks by a user, under an assumed user-behavior model. Such models are known as click models. They are common in learning to rank (Agichtein et al. 2006; Richardson, Dominowska, and Ragno 2007; Craswell et al. 2008; Chuklin, Markov, and de Rijke 2015) and have also been used to learn the best ranked list online from observed clicks (Kveton et al. 2015).

Recall that \( s(i,j) \in [0, 1] \) is the item-topic relevance and that \( w_j \in [0, 1] \) is the topic preference. We consider the following random process for generating clicks on items, for any list \( I \):

- For each topic \( j \), topic attraction is sampled as \( W_j \sim \text{Ber}(w_j) \). The random variable \( W_j \) denotes that the user is interested in topic \( j \).
- For each relevance \( s(i,j) \), item-topic attraction is sampled as \( S_{i,j} \sim \text{Ber}(s(i,j)) \). The random variable \( S_{i,j} \) denotes that the user would be interested in item \( i \) when interested in topic \( j \). We assume that \( S_{i,j} \) are sampled independently across the items and from \( W_j \).
- For any list \( I \), the user scans it and clicks on the first attractive item in each topic that they are interested in. This objective can be written as

\[
    F(I) = \sum_{j \in \mathcal{J}} W_j \left[ \prod_{i \in I} (1 - S_{i,j}) \right]
\]

The above is a generalization of the cascade model (Richardson, Dominowska, and Ragno 2007; Craswell et al. 2008) to multiple topics of interest. We can show that \( f_{\text{cov}}(I) = E[F(I)] \) from assuming that \( S_{i,j} \) are sampled independently across the items and from \( W_j \). In particular, we would get

\[
    E[W_j \prod_{i \in I} (1 - S_{i,j})] = w_j \prod_{i \in I} (1 - s(i,j))
\]

The rest of the derivation follows from the linearity of expectation.

**Experiments**

In this section we describe the evaluation of our proposed model on research advisor and movie recommendations. In both use cases we followed the same evaluation protocol to ensure the consistency between the results.

**Experimental Setting**

In our experiments, we define a simulated user as follow: each user is interested in a set of topics. Each topic of interest in this set \( j \in \mathcal{J} \) is represented by a positive weight \( w_j > 0 \) in the User Profile (while the rest of the topics have weight \( w_j = 0 \)). Weights are assigned randomly to each \( \text{(user, topic)} \) pair and represent the preference of the user for topic \( j \). The item-topic relevance score \( s(i,j) \) is inferred from the data and pre-calculated for every \( \text{(item, topic)} \) pair in advance. In every run of the simulation, we generated two sets of recommendations using models described in Equation (1) and Equation (4). We utilized the first set as the baseline (Top-K) model and the second one as proposed (Coverage) model.

While several metrics for measuring recommendation diversity were proposed (Vargas and Castells 2011), there is no standard metrics focused on coverage. In our study, we introduce **Satisfactory Index** as a coverage metric. We define the **Satisfactory Index** as the ranking position in the list which the user exploring recommendations sequentially has to reach to find at least one **Satisfactory Item** for each topic \( j \) in the User Profile. A **Satisfactory Item** for a topic is defined as an item with higher than average item-topic relevance score \( s(i,j) \) for this topic. **Satisfactory Index** combines a standard first-click rank metrics that assesses how high a recommender system places first relevant item in the ranked list with a worst-case scenario where the true topic of interest is positioned last in the list.

**Conditions**

We compared the two models varying three sets of simulation parameters. First, we varied the number of topics in the
User Profile in the range from two to ten to cover a wide range of possible real-life scenarios. Second, we fixed the number of items in the User Profile and varied the number of simulation instances (sample size) from 100 to 1000 to observe the stability of our proposed approach in larger use cases. Finally, we ran independent simulations for two ways to represent user interest for a topic in the User Profile. In one of the simulations we assigned a fixed weight (1) to all the topics of interests and in the other one we set the weight to a random positive number between 0 and 1.

Use Case 1: Research Advisor Recommendation

Dataset In the first use case, we employ a local dataset of research advisors $i$ and their relative expertise level to a set of research topics $j$ and run simulated scenarios. This dataset has been utilized by Grapevine system that is fully described in (Rahdari, Brusilovsky, and Babichenko 2020).

For the purpose of this experiment we utilized a portion of the data set that describes the relationship between research advisors and topics. We used the latest version of the dataset with 172 research advisors and 16,350 research topics. We perform 1000 rounds of simulation per experiment condition (No. of Topics in the User Profile) was assigned at random.

Results In this section we describe the results of our offline experiments in the use case research advisor recommendation. Figure shows the results of our experiments in the first two conditions described above. Figure 1(a) displays the average satisfactory index in a range of different number of topics in the User Profile. As is evident from the figure, the Coverage model produced a lower average satisfactory ranking across all different topic sizes. The green line indicates that on average, the Coverage model could satisfy the experiment condition (covering all the topics with at least one Satisfactory Item) in fewer steps than the number of topics in the User Profile. This ratio is almost double in the case of Top-K model.

As demonstrated in Figure 1(b), for Coverage model the different sample sizes have a minimal effect on the average satisfactory ranking. This effect appears to be slightly fluctuating when Top-K model was used to generate the recommendations.

Finally, Figure 4 compares the two models in regard to user’s relative interest to each topic in the User Profile. There are not apparent difference between the results under these two conditions. However when relative weights exists, the Coverage seems to perform slightly better.

Use Case 2: Movie Recommendation

We replicated our experiments in a different context and by using a more standard dataset to reduce the potential biases caused by using a local data set. Despite the changes in the data, all the other experimental conditions remained the
same. Two main variations between movie recommendation and advisors recommendation include the way we define the item-topic relevance score and the ratio between topics and items. In the advisor recommendation use case the item-topic relevance score pre existed in the dataset and there are considerably more topics (research interest) than items (research advisor) in the dataset. In movie recommendation use case we calculated the relevancy score between each movie and genre using the user rating data and the number of items (movies) exceed the number of topics (genres).

Dataset In this context, we used MovieLens 100K Dataset (Harper and Konstan 2015) with 100,000 ratings and 3,600 tag applications applied to 9,000 movies in 19 genres by 600 users. Similar to the first use case, we performed various simulations in various conditions including different topic sizes (from two to ten) in the User Profile and different number of simulations (from 100 to 1000) for a fix sets of conditions. We also include the condition where each genre in the User Profile is weighted with a number between 0 and 1.

Results In this section, we present the results of our offline evaluation using the MovieLens dataset. Figure 3a and Figure 3b illustrate the comparison between the two models based on different number of genres in the User Profile and different number of sample size (the number of times we ran the simulation for a specific condition) respectively. This result is comparable with the similar results from the first use case. However, the difference between the two models seems to be more prominent in the case of movie recommendation.

Similar to the previous use case, assigning weights to the each topic in the User Profile did not result in any significant changes in the performance. The Coverage algorithm still outperforms the Top-K model in both cases.
Performance
In order to demonstrate the effectiveness of our proposed approach in a practical setting, we compare the performance of the baseline and Coverage model. This benchmark (Figure 5) shows the average time (in millisecond) that each model takes to compute one instance of the simulation. This test is being done with a fixed number of topics in the User Profile with a fixed weight associated to each topic. The results show a relatively comparable performance outcome for the two model. However, Top-K model had a sub-standard performance in movie recommendation use case.

Conclusions
In this paper we describe a common problem with top-k ranking model for the recommendation and proposed a solution that increases the coverage of relevant items in an interactive recommendation setting. Our proposed ranking approach attempts to balance traditional relevance with coverage providing the users a better opportunity to explore the items that usually being overlooked due to using a modular top-k rating model.

We ran simulated use cases related to two problem of advisor and movie recommendations. The results of evaluation indicated that our approach considerably increases chances that the user finds relevant items in the first step of the recommendation dialog. Although we do not simulate the rest of the dialog, we believe that it would also improve due to the improvement in the first step. We leave a comprehensive study of this improvement for future work.

References
Adomavicius, G.; Sankaranarayanan, R.; Sen, S.; and Tuzhilin, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems 23(1):103–145.
Adomavicius, G.; Mobasher, B.; Ricci, F.; and Tuzhilin, A. 2011. Context-aware recommender systems. AI Magazine 32(3):67–80.
Agichtein, E.; Brill, E.; Dumais, S.; and Ragno, R. 2006. Learning user interaction models for predicting web search result preferences. In 29th ACM SIGIR Conference, 3–10.
Bostandjiev, S.; O’Donovan, J.; and Höllerer, T. 2012. Tasteweights: A visual interactive hybrid recommender system. In 6th ACM Conference on Recommender System, 35–42.
Burke, R.; Hammond, K.; and Young, B. C. 1997. The findme approach to assisted browsing. IEEE Intelligent Systems 12(4):32–40.
Carbonell, J., and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st ACM SIGIR Conference, 335–336.
Chen, L., and Pu, P. 2012. Critiquing-based recommenders: survey and emerging trends. User Modeling and User-Adapted Interaction 22(1):125–150.
Chuklin, A.; Markov, I.; and de Rijke, M. 2015. Click Models for Web Search. Morgan & Claypool Publishers.
Craswell, N.; Zoeter, O.; Taylor, M.; and Ramsey, B. 2008. An experimental comparison of click position-bias models. In Proceedings of the 1st ACM International Conference on Web Search and Data Mining, 87–94.
Cremonesi, P.; Koren, Y.; and Turrin, R. 2010. Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the Fourth ACM Conference on Recommender Systems, 39–46.
Edmonds, J. 1970. Submodular functions, matroids, and certain polyhedra. In Combinatorial Structures and Their Applications: Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications. 69–87.
El-Arini, K.; Veda, G.; Shahaf, D.; and Guestrin, C. 2009. Turning down the noise in the blogosphere. In 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 289–298.
Harper, F. M., and Konstan, J. A. 2015. The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. 5(4).
He, C.; Parra, D.; and Verbert, K. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. Expert Systems with Applications 56(C):59–27.
Jambor, T., and Wang, J. 2010. Optimizing multiple objectives in collaborative filtering. In 2010 ACM conference on Recommender systems, 55–62. ACM.
Kveton, B.; Szepesvari, C.; Wen, Z.; and Ashkan, A. 2015. Cascading bandits: Learning to rank in the cascade model. In 32nd International Conference on Machine Learning.
Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions - I. Mathematical Programming 14(1):265–294.
Parra, D., and Brusilovsky, P. 2015. User-controllable personalization: A case study with setfusion. International Journal of Human-Computer Studies 78:43–67.
Rahdari, B.; Brusilovsky, P.; and Babichenko, D. 2020. Personalizing information exploration with an open user model. In 31st ACM Conference on Hypertext and Social Media, 167–176. ACM.
Richardson, M.; Dominowska, E.; and Ragno, R. 2007. Predicting clicks: Estimating the click-through rate for new ads. In Proceedings of the 16th International Conference on World Wide Web, 521–530.
Smyth, B.; McGinty, L.; Reilly, J.; and McCarthy, K. 2004. Compound critiques for conversational recommender systems. In IEEE/WIC/ACM International Conference on Web Intelligence (WI’04). IEEE.
Vargas, S., and Castells, P. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In Fifth ACM Conference on Recommender Systems, 109–116.
Ziegler, C. N.; McNee, S. M.; Konstan, J.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In 14th international conference on World Wide Web, 25–32. ACM Press.