Diverse User Preference Elicitation with Multi-Armed Bandits

Javier Parapar*
javier.parapar@udc.es
Universidade da Coruña
A Coruña, Spain

Filip Radlinski
filiprad@google.com
Google Research
London, United Kingdom

ABSTRACT

Personalized recommender systems rely on knowledge of user preferences to produce recommendations. While those preferences are often obtained from past user interactions with the recommendation catalog, in some situations such observations are insufficient or unavailable. The most widely studied case is with new users, although other similar situations arise where explicit preference elicitation is valuable. At the same time, a seemingly disparate challenge is that there is a well known popularity bias in many algorithmic approaches to recommender systems. The most common way of addressing this challenge is diversification, which tends to be applied to the output of a recommender algorithm, prior to items being presented to users.

We tie these two problems together, showing a tight relationship. Our results show that popularity bias in *preference elicitation* contributes to popularity bias in *recommendation*. In particular, most elicitation methods directly optimize only for the relevance of recommendations that would result from collected preferences. This focus on recommendation accuracy biases the preferences collected. We demonstrate how diversification can instead be applied directly at elicitation time. Our model diversifies the preferences elicited using Multi-Armed Bandits, a classical exploration-exploitation framework from reinforcement learning. This leads to a broader understanding of users' preferences, and improved diversity and serendipity of recommendations, without necessitating post-hoc debiasing corrections.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

recommender systems, preference elicitation, diversity, bandits

ACM Reference Format:
Javier Parapar and Filip Radlinski. 2021. Diverse User Preference Elicitation with Multi-Armed Bandits. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441786

*Work carried out as Visiting Faculty Researcher at Google.

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441786

1 INTRODUCTION

Recommender Systems (RSs) [46] have become ubiquitous for recommending movies [19], music [54], books [57], news [34], and in numerous other domains. Their goal is to help users more quickly find relevant items in vast catalogues.

One of the most common approaches is Collaborative Filtering (CF), which depends on rich user preference information to discover relationships between items, and in turn allows user-user and item-item similarity to be estimated. Given a user profile, items of interest to similar users can then be suggested. Preferences can be explicit (for example, ratings on books, or likes of posts) or implicit (for example, the history of songs a user has listened to, or films the user has watched). However, for CF approaches to work well, algorithms need sufficient user preferences, which are not always available. In fact, cold start is one of the most acute problems of CF methods and is heavily studied [50]. *User cold start* refers to the problem of new users, where a system has no or little preference information. In this situation, the usual solution to enable CF algorithms is to perform preference elicitation: The system asks the user for some initial preference information (e.g., [14, 44, 45]).

However, elicitation is useful beyond the cold start problem. For instance user interests change or conflict over time [42], or depend on context [7]. Beyond this, systems can improve by validating preferences collected previously, to remove noisy observations [4, 53, 58]. In general terms, an elicitation algorithm's objective is to decide which items or questions to ask a user so as to obtain information about their preferences most effectively.

Recommendation accuracy has dominated RSs research in the past. However, there has been a recent recognition that other aspects are also important. For instance, item novelty and recommendation diversity have been found to improve user satisfaction [11, 22, 37]. As such, it is now generally accepted that recommendation systems should help users find items they would not have discovered otherwise. This has led to studies measuring unexpected items, or *serendipity* [2, 29], which is often associated with user satisfaction [16]. Indeed, it has been shown that improving diversity, as a proxy for serendipity, also improves user satisfaction [10].

To improve such aspects, the most common approach is to improve diversity in recommendation results, as surveyed by [31].

Diversification algorithms reorder the items scored for relevance to increase item diversity. However, the best performing RSs frequently rank very similar items together at the top of the ranking (as similar items are of similar relevance). Diversification thus tends to delve deep into the recommendation list, increasing the probability of selecting non-relevant items to recommend. This creates a trade-off between accuracy and diversity [38].

In this work, we present an alternative approach, addressing the diversity problem at a fundamentally different stage: We introduce

a method for the diversification of the *preference elicitation* process. By tackling diversity when user interests are established, our method provides the recommender algorithm with a broader and more complete user profile. This variety of user interests allows the recommendation algorithm to produce recommendations less prone to popularity bias and overspecialization. Specifically, we propose a Multi-Armed Bandits based algorithm [26] that improves both thematic and item diversity during preference elicitation.

We make two key contributions:

- We present the first model that diversifies preference elicitation. This approach is independent of any particular recommendation algorithm, and results in broader user profiles.
- We present a new methodology for offline evaluation of preference elicitation, reducing an important bias that we identify in existing approaches, favouring apparent accuracy of greedy-like algorithms.

Next, we discuss related work and provide background on key concepts. Then Section 3 presents our algorithm for diversified preference elicitation. We describe the experimental setup in Section 4 and results in Section 5. Finally, we conclude in Section 6.

2 RELATED WORK

To the best of our knowledge, this is the first attempt to diversify the preference elicitation process. However, there is extensive work on preference elicitation, diversification of recommendation lists, and Multi-Armed Bandits for recommendation. We now summarize this previous work and introduce some of the key relevant concepts.

2.1 Preference Elicitation

Preference elicitation refers to acquiring explicit information from a user regarding their tastes about items, with the goal of building a user profile [14]. This elicitation process is useful in many scenarios: For users for which the RS has no information at all (new or *cold-start* users); for users with changing preferences (preference revision); and for users whose preferences have been estimated by the system (usually implicitly by observing user actions) and which require validation [4, 53] (preference validation).

In this paper, we will address only the algorithmic process of selecting the items to be presented to the user. In particular, we restrict ourselves to absolute preference data (preference about a single item). Although there are some advantages of eliciting preferences between pairs of items [25, 41], we do not address a pairwise formulation of our model here. Pertinently, Christakopoulou et al. [13] addressed interactive preference elicitation for restaurant recommendation, proposing models for both absolute and relative preferences. In their setting, they found that absolute models perform better than the relative ones, and show a greedy algorithm performs best in terms of optimizing Average Precision.

Most previous work in this area optimizes recommendation accuracy based on the user preference obtained. For instance, Salimans et al. [48] proposed a Bayesian factor model showing how it can be used for active relative preference elicitation in an active learning fashion. The model reduces the entropy of the posterior distribution. Rashid et al. [44] recommended the integration of popularity and entropy in the selection criteria, and this was reiterated in [17]. The advantage of this (rather than only pursuing entropy reduction) is that popular items have a higher chance of having been rated in *offline test data* used to evaluate elicitation algorithms. This fact, combined with the tendency of popularity-biased user profiles to produce popularity-biased recommendations, suggests that those recommendation lists would tend to show higher precision in offline evaluation. This argument resulted in the appearance of models designed to work around two well-known problems of offline RSs evaluation: popularity bias, and missing ratings. Rashid et al. [45] also recognized the problem of the missing item ratings and present an entropy variant for that problem. Their *Entropy*0 method presented better accuracy figures than popularity in isolation. Similarly, Sepliarskaia et al. [51] presented an approach to create a fixed preference questionnaire offline instead of interactively computing the next item to be presented after each answer. Static questionnaires facilitate the logistics of the preference elicitation without the need of an online feedback loop-back. All this work contrasts with optimizing for *user satisfaction*.

2.2 Results Diversification

Diversity in Recommendations has attracted attention as well [6], with much other work surveyed by Kunaver and Porl [31]. In the same way that top-n recommendation has benefited from advances in search ranking, diversity in recommendation also takes ideas from Web search result diversification. In particular, the Maximal Marginal Relevance method (MMR) presented by Carbonell and Goldstein [9] has been the base of many RS diversification proposals. In general terms, such approaches try to maximize a sub-modular function that combines both items' estimated relevance and dissimilarity with already selected items.

Diversity is commonly considered together with another desired property, namely novelty [11]. Although strongly related, these concepts refer to subtly different aspects beyond accuracy. Novelty refers to how new an item is *for a user*, given what the user has already seen. Diversity, on the other hand, is applied over a set of items and refers to how different the items are. In this regard, diversity has two sides: On the item supply side, diversity is about which items from the catalog are recommended [15]. On the user side, diversity is considered in individual's recommendation lists: Are items repetitive, or do they provide a broad range of choices?

Arguably, one of the counterparts of diversity is popularity bias [23]. This bias refers to the fact that frequently rated items often dominate recommendations. Although the recommendation of popular items may benefit the *accuracy* of an RS, promoting such items tends to reduce diversity. Several works have studied how RSs tend to narrow recommendations gradually [1, 30], noticing that these tend to be more biased towards popularity than user profiles.

Another closely related concept is unexpectedness. This refers to the recommendation of items that are ranked at a position that is significantly above that expected for a random user. This impacts the user's perceived value of the recommendation: "Is the system recommending items that I would not find otherwise?". Many user studies have explored the effect of unexpected items on user satisfaction. For instance, Castagnos et al. [10] show how diverse recommendations may reduce precision but still help to increase users' satisfaction.

More recently, Chen et al. [12] demonstrated significant causal relationship from serendipity to user satisfaction and purchase intention in a large-scale user survey (3.000 users).

Adamopoulos and Tuzhilin [2] presented an analysis of the relationship between unexpectedness and diversity, proposing a utility function for calculating the usefulness of recommendations. In the past, many algorithms were presented to improve the diversity of the recommendations. For instance, Abdollahpouri et al. [1] introduced a regularization factor in the RankALS recommendation algorithm. Its objective is to diversify the recommendations by including medium-tail items. However, most existing work addresses diversification at the item level. Ziegler et al. [59] presented the first attempt at producing diverse recommendations at a topic level (i.e. types or categories of items): their model considers the different types of items through the use of a taxonomy-based similarity metric. In fact, Rong and Pearl [47] studied the importance of topical diversity with respect to item diversity. In that work, perceived topical diversity is found to have a greater effect on perceived value and ease of use of the RS.

2.3 Multi-Armed Bandits

Multi-Armed Bandits [26] (MABs) are a well-known reinforcement learning framework. They are modeled on a gambler trying to choose which slot machine to play, whose objective is to maximize total return after t pulls (trials), by modeling the probability of each machine providing a payoff. Each machine (arm) can be pulled, and it emits a reward. After that the reward is observed, the estimate of the utility of pulling that arm can be recalculated. At any time step, the gambler can decide to *exploit* the machine that has been most profitable in the past, or *explore* new machines.

Some authors have considered using MABs for item recommendation. Li et al. [32] present LinUCB, a variant of the UCB (Upper Confidence Bound) bandit algorithm for contextualized news recommendation. Bayesian bandits have also been used for online recommendation. Kawale et al. [27] presented the use of Thompson Sampling in an online recommendation setting for selecting items with a matrix factorization recommendation method. Another application of MABs in RSs is clustering and neighborhooding. Li et al. [33] present the use of MABs for adaptive clustering in content recommendation based on exploration-exploitation strategies. More recently, Sanz-Cruzado et al. [49] presented how to use MABs for producing a variant of the classical kNN recommender. They propose the use of the exploratory nature of the MABs for selecting neighbors with Thompson-Sampling.

More related to our work, MABs have also been used for preference elicitation. Kohli et al. [28] presented a stochastic multi-armed bandit for exploring user preferences online, with an application to news articles, while minimizing user abandonment. More recently, Christakopoulou et al. [13] presented several bandit methods such as Thompson Sampling and UCB for the elicitation task.

Other ranking tasks have also benefited from the use of MABs. For instance, Losada et al. [35, 36] presented their use for pooling in the construction of evaluation datasets for Information Retrieval. On the other hand, Radlinski et al. [43] use MABs for learning diverse rankings of web documents based on user clicks, contrasting with greedy ranking approaches that show redundant information.

3 DIVERSIFICATION OF THE ELICITATION PROCESS WITH MULTI-ARMED BANDITS

When using an RS, a feedback-loop exists between ratings collected and future recommendations [40]. If items are recommended without considering diversity, recommendations may unintentionally become less diverse over time. This can be counteracted by ensuring long-tail items are present in recommendation lists to obtain a fuller understanding of users' preferences.

The lack of diverse items is exacerbated when we address preference elicitation with classical active learning approaches. By using classical algorithms, we reduce the variety of choices in the elicitation process by driving the elicited preferences away from representing the full breadth of a user's interests. Systems that use active learning to explore the user's profile must explicitly optimize exploring more diverse items, as these are items that the user is less likely to have rated. Exploit-only approaches lead to popular items in questions, resulting in the RS missing the users' more varied and distinctive preferences. In this context, diversification during the elicitation process aims to maximize the informativeness of elicited items for a broader understanding of the user.

3.1 Diverse Preference Elicitation

While past work has shown that entropy-reduction techniques are often best for recommendation accuracy [14], as discussed above there is also value in presenting diverse options. We hypothesize that having an elicited profile that is diverse in user's topics and tastes would help the subsequent recommendation algorithm to produce more diverse results, and avoid popularity bias inherent to most of them. With that goal in mind, we present DPE, a Diversified Preference Elicitation model based on Multi-Armed Bandits. In this context, *diversity* refers to the *user's* perceived diversity. It can be considered at two levels. First, item diversity: how varied the items presented to the user are among themselves. Second, topic diversity: how the items presented are topically different, i.e. from different categories of interests. Both aspects are important to have a better understanding of user preferences. Therefore, we include both as objectives for our preference elicitation algorithm. As previously commented, we will work here with absolute preferences, leaving the formulation of a pairwise variant to future work. Specifically, the user will be presented with one item at a time, being asked to express their absolute preference over it.

In particular, we propose the use of Thompson Sampling Multi-Armed Bandits with Gaussian Priors [3]. Algorithm 1 presents the outline of our model, which works as follows. For each user u (in the set of all users $\mathcal{U}$) we run a different bandit. In our bandit, each arm $a \in \mathcal{A}$ represents an item topic. An item topic can be defined differently depending on the application domain; it can correspond with genres of movies, types of food cuisines in restaurant recommendation, or types of items in an e-commerce context. Moreover, an item could belong to more than one topic, e.g., a movie being both adventure and sci-fi. The arms are modeled by Normal Distributions whose parameters we initialize equally [24].

The elicitation process consists of running as many time steps t as desired elicited items e. In each iteration, an arm is selected by sampling from the Gaussian distributions of each topic, and selecting that (*next_topic*) which has highest sampled value. By

Algorithm 1: Diverse Preference Elicitation (DPE)

forall $u \in \mathcal{U}$ **do**
  forall $a \in \mathcal{A}$ **do**
    $\mu_a(0) \leftarrow 0$;
    $k_a(0) \leftarrow 0$;
  end
  $P[u] \leftarrow \{\}$;
  foreach $t = 1, 2, ..., e$ **do**
    forall $a \in \mathcal{A}$ **do**
      Draw sample θ_a from $\mathcal{N}(\mu_a(t-1), \frac{1}{k_a(t-1)+1})$;
    end
    $next\_topic \leftarrow \arg\max_a \theta_a$;
    forall $i \in next\_topic$ **do**
      $\delta_i \leftarrow score^{div}(i, P[u])$;
    end
    $next\_item \leftarrow \arg\max_i \delta_i$;
    $r(t) \leftarrow reward_f(next\_item, r(u, next\_item))$;
    forall $a \mid next\_item \in a$ **do**
      $\mu_a(t) \leftarrow \frac{\mu_a(t-1) \cdot k_a(t-1) + r(t)}{k_a(t-1)+2}$;
      $k_a(t) \leftarrow k_a(t-1) + 1$;
    end
    $P[u].add(next\_item)$;
  end
end

selecting an arm, the bandit is deciding which topic to ask preferences over. With a topic having been selected, an item belonging to that topic is selected as the elicitation item (*next_item*). To select *next_item*, all items i in that topic are ranked according to how diverse they are with respect to the existing user profile $P[u]$. This ranking is a second diversification component, promoting item-to-item diversity in the elicitation process. In the algorithm, $score^{div}(i, P)$ can be selected from alternatives that we present next. After obtaining a preference from the user for the selected item, a reward $r(t) \in \{0, 1\}$ is computed and every arm a (topic) to which *next_item* belongs is updated by computing the posterior $Pr(\tilde{\mu}_a|r(t)) \propto Pr(r(t)|\tilde{\mu}_a) \cdot Pr(\tilde{\mu}_a)$. In our experiments, we use the normalized rating value as the reward, but other reward functions could be explored. In the case of our algorithm, the bandit's exploration can be adjusted by tuning the initial value of $k_a(0)$: The higher this value, the lower the variance of the Gaussian distribution and, therefore, the less exploration the algorithm would perform. Our hypothesis is that the MAB framework would naturally cope with *exploitation* of topics that the user likes the most versus *exploration* among all possible tastes, obtaining a broader and more complete user profile. Moreover, we complement the thematic exploration with the item-to-item diversification to reduce redundancy in the user profile, obtaining more value from answers.

3.2 Arm ranking strategies

For selecting an item once we pull a topic, we rank all items in the arm and pick the first. The sorting criterion can be chosen differently. Here we use it to promote item-to-item diversity of elicited items. However, the potential to use other arm ranking policies is that they can be adjusted to any elicitation scenario. For instance, given cold-start users without any existing background knowledge, popular items naturally have a higher chance of being recognized by the user. Under our framework, we may use item popularity as an arm ranking strategy to get a topically diverse preference seed set, and then flip to another policy to promote item-to-item diversification. On the other hand, if we are in a preference validation setting, the implicit preferences inferred by the system can be used to rank the items. In preference revision, we could choose to prioritize older preferences to be revisited. Next, we present three scoring strategies:

3.2.1 Hellinger distance. This symmetric f-divergence is used to compute similarities between probability distributions in terms of the Hellinger integral [21]. It is the probabilistic equivalent to the Euclidean distance:

$$score^{Hellinger}(i, P[u]) = \frac{1}{|P[u]|} \sum_{j \in P[u]} \frac{1}{\sqrt{2}} \sqrt{\sum_{v \in \mathcal{U}} (r(v, i) - r(v, j))^2} \quad (1)$$

3.2.2 Intra-List Distance. ILD is a distance based on intra-list similarity [59] that tries to capture the diversity of a list. ILD measures the cosine distance of the new item to every existing item, averaging over the size of the user's profile. The higher the ILD value, the more profile diversity would be contributed by a new item:

$$score^{ILD}(i, P[u]) = \frac{1}{|P[u]|} \sum_{j \in P[u]} 1 - \frac{\sum_{v \in \mathcal{U}} r(v, i) \cdot r(v, j)}{\sqrt{\sum_{v \in \mathcal{U}} r(v, i)^2} \cdot \sqrt{\sum_{v \in \mathcal{U}} r(v, j)^2}} \quad (2)$$

3.2.3 Oracle. This is an oracle arm sorting strategy that ranks the items according to the actual rating that the user would give the item (independently of elicited items). Knowing this rating beforehand would only be possible for validation settings over inferred preferences. However, we present this method as an oracle-based strategy, not pursuing item-to-item diversification, to see a non-diversified upper-bound of topic-only diversification:

$$score^{Oracle}(i, P[u]) = r(u, i) \quad (3)$$

4 EXPERIMENTAL METHODOLOGY

In this section, we present our experimental design. In particular, we answer to the following research questions:

- RQ1) Does diversification of the elicitation process reduce recommendation accuracy when compared to existing accuracy-optimized recommendation methods?
- RQ2) Does diversification of the elicitation process produce more diverse and unexpected recommendations?

By answering these questions we will have a better understanding of the merits of diversifying the elicitation process. In particular, we will analyse the trade-off between accuracy and diversity when diversifying at an earlier stage in the recommendation process.

4.1 Datasets

One of the objectives of this study is to assess the effect of topical diversification. We evaluate our approach on two widely used datasets where topic information is available: The *Movielens 20M* dataset [19] relying on movie genre information, and the *Amazon*

Product Review 2014 dataset [20] where items are classified by type on the e-commerce platform. Both datasets have items rated by users with graded preferences, and every item may belong to one or more categories. Following common practice, we filter out users with few ratings — in this case, removing those with fewer than 100 ratings, given that our evaluation elicits up to 75 preferences per user. We also note that as we use the same recommender algorithm for all elicitation strategies, this filtering affects all methods equally. Table 1 presents a summary of the characteristic of the final collections. Importantly, these datasets are quite different, not only in domain but also in terms of sparsity: There are on average 600 ratings per item in the Movielens dataset, while only 3.3 in the Amazon dataset, the latter being much more challenging for RSs.

Table 1: Summary of the statistics of the datasets.

Dataset	Users	Items	Ratings	Cat.	$\frac{\text{Ratings}}{\text{Users}}$	$\frac{\text{Ratings}}{\text{Items}}$	$\frac{\text{Cat.}}{\text{Items}}$
Movielens	51,869	26,654	15,970,206	20	308	600	2.04
Amazon	29,598	2,018,050	6,786,371	25	229	3.3	1.09

4.2 Evaluation Methodology

One important challenge in evaluating preference elicitation algorithms is how to obtain item ratings from users. Following common practice, we use an offline evaluation protocol based on naturally acquired ratings with known but held-out user profiles. This allows us to study how the number of questions asked of each user affects performance, and report it for any particular budget of questions answered. We vary the number of elicited items from 5 to 75. The actual preferences are obtained directly from the held-out ratings from the user. Specifically, the procedure we follow is:

(1) At each time step, the elicitation algorithm takes as input the training data (i.e. the items in the training split for the user), and any answers previously collected.
(2) The recommendation algorithm produces a recommendation list for the user based on the output of the elicitation process.
(3) The quality of the recommendations is evaluated using different metrics allowing us to plot the curves of performance for different numbers of elicited items.
(4) Then, the elicitation algorithm decides the next item to present to the user as a preference elicitation question.

We restrict our experiments to use the same recommendation algorithm with all the elicitation methods. In this way, we isolate the merits of the elicitation models without considering the influence of the chosen recommendation algorithms. For the recommender, we use the recent WSR algorithm [56]. The recommendation algorithm is tasked with ranking all items in the test split for which there is no rating on the training split for the user. Comparing performance of different elicitation strategies using a single, consistent and reasonably strong recommender algorithm allows us to robustly separate the recommender algorithm from the preference data upon which recommendations are based. We expect similar relative performance given any recommender algorithm.

4.2.1 Standard Evaluation. Our first evaluation follows the *TestItems* approach [5]. We produce an 80%-20% training-test split, taking 80% of each user's ratings as training for running the elicitation process and 20% for evaluation. Importantly, the elicitation algorithms are blind to the non-elicited preferences for each given user, even when those are present in the training set. That is, for example, when computing the neighbors of the given user, the algorithm would only see the ratings of the user over items already elicited, not her full training profile. The WSR has a single parameter, the neighbour size parameter k. Following standard practice [52, 55], for each algorithm we select the optimal value of $k \in \{5, 10, 20, \ldots, 100\}$ that maximizes overall *nDCG*@100 performance (see Section 4.4) in terms of total area under the curve.

We observe that this evaluation approach has both benefits and drawbacks: By mimicking online preference elicitation based on existing user data, *TestItems* provides the ability to reliably repeat the elicitation with many experimental variants at minimal cost, and is thus commonly used [13, 51]. On the other hand, any offline evaluation protocol lacks some realism in that it only allows answers over items previously rated by the user [18]. This scenario matches that of preference validation and preference revision tasks. As we are using the same methodology for all methods, the relative merits of the elicitation algorithms are compared fairly. We also observe that this limitation is likely to affect *absolute values* of the metrics for each method, therefore absolute values reported should be considered with caution. However, our goal is to evaluate the relative performance of the algorithms compared.

4.2.2 AvailablePreferences: Evaluation with Leftovers. One of the key weaknesses of the standard TestItems approach, when used for preference elicitation, is that many items in the training split are never used: The split contains answers to questions the elicitation system *may* ask. If the system does not *choose to* ask the user about any particular item, the rating for that item is, in effect, wasted data — we term such unused rated items *leftovers*.

This leads us to a variant of the TestItems approach that we term *AvailablePreferences*. We start by observing that leftovers still indicate user preferences. As mentioned earlier, recent work has shown that missing ratings strongly affect the robustness of offline evaluation [55]. While robust metrics and deeper cut-offs mitigate the effect of missing ratings, for evaluating preference elicitation algorithms it is possible to use leftovers for comparison between different methods asking *the same number of questions*. In particular, we propose to compute the results for the different metrics by including the leftover preferences as relevance judgments, together with those in the test set used by TestItems. We also remark that this does not affect the recommendation process: the exact same system output is produced under both evaluation methodologies.

4.3 Baselines

Although the diversification of the elicitation process has not been addressed before to the best of our knowledge, we constructed baselines as variants of existing approaches for result diversification.

As a first baseline we use the Greedy method from [13]. In that work, the authors compare several algorithms for non-diversified preference elicitation, finding that two algorithms (UCB and Greedy) performing best depending on the number of elicitation questions. We use the simpler **Greedy** approach as a baseline: At each time step, it solicits the relevance of the item with highest estimated

relevance by a recommendation algorithm that uses the previously elicited items, i.e., the top item in the recommendation list.

As a second baseline, we adapt a standard post-hoc diversification, namely **MMR** [9]. Our adapted version, Greedy+MMR, diversifies recommendations generated by WSR over the elicited profiles obtained by the Greedy method. For MMR, we used $\alpha = 0.5$, which equally weights diversity and relevance. One could modify α to favor either relevance or diversity. If we were to use $\alpha = 1$, this would correspond to the Greedy baseline. Alternatively, we could opt for optimizing diversity, which would reduce accuracy further, as we will see in the results.

4.4 Evaluation Metrics

Following the advice from [55], we use Precision and Normalised Discounted Cumulative Gain (nDCG) for recommendation accuracy. Further, we present two metrics for item and topic diversity.

Precision@k measures the percentage of relevant items in the top-k recommendations for the user (L_u^k). We consider relevant items for the user ($\mathcal{R}_u$) as those with ratings at least 4. We report $P@k$, the averaged value across all the users.

$$P_u@k = \frac{|L_u^k \cap \mathcal{R}_u|}{k} \tag{4}$$

nDCG@k uses graded relevance (rating values), weighting items by their position p in the recommendation list, with discounting function $D(p)$. It is formulated as follows:

$$nDCG_u@k = \frac{\sum_{p=1}^{k} G(u,k,p)D(p)}{\sum_{p=1}^{k} IG(u,k,p)D(p)} \tag{5}$$

where $G(u,k,p)$ is the gain obtained by recommending the item $L_u^k[p]$ to the user u and $IG(u,k,p)$ is the maximum possible gain at that position for the ideal ranking of size k. As in the case of precision, we report $nDCG@k$ averaged across all users. We used the `trec_eval` implementation for computing Precision and nDCG with the default discount function and the ratings as gain values.

Serendipity@k. For measuring diversity, this metric reflects popularity bias, and evaluates how the recommendations provide value to the user. It is related to the unexpectedness metric formulated in [39], and to Konstan and Ekstrand's formulation[1]. Intuitively, the metric reflects how unexpected it is to find an item in the recommendation list for a user. It is computed by counting how much more probable it is to find item i at position p of the recommendation list (L_u) for the user u relative to how probable it is to find that item for a random user.

$$serendipity_u@k = \frac{1}{k}\left[\sum_{p=1}^{k} max\left(Pr_{L_u^k[p]}(u) - Pr_{L_u^k[p]}(\mathcal{U}), 0\right) \cdot r(u,i)\right] \tag{6}$$

Here $Pr_i(u) = \frac{k - rank_i}{k-1}$, i.e., the probability of an item i for user u is proportionally inverse to the position of the item in the recommendation list. The metric compares how the probability differs from the overall probability for all the users: $Pr_i(\mathcal{U}) = \frac{\sum_{u \in \mathcal{U}} Pr_i(u)}{|\mathcal{U}|}$. For instance, a non-personalized recommendation strategy based on popularity alone would have a score of 0. Moreover, as we weight the difference in probability by the rating that the user assigns to the item ($r(u,i)$), a method that recommends diverse non-relevant items, even if they are long-tail items, would also score 0. There are alternative definitions for serendipity [16], although they require assumptions on expected recommendations, and these assumptions could cause the metric to be biased to specific approaches. We report *serendipity@k* averaged over all users.

Topdiv@k. We also define a new metric for measuring how *topically* diverse a recommendation list is. As not every user in the collection has the same degree of interest in different categories, we measure this topical diversity with respect to each user's preferences. For doing so, we measure the Kendall's τ rank correlation between the distribution of user topical interests on the rating matrix and its distribution on the recommendation list:

$$topdiv_u@k = \tau(\mathcal{T}(\mathcal{R}_u), \mathcal{T}(L_u^k)) \tag{7}$$

where $\mathcal{T}(X)$ is the probability distribution of relevant items on the categories. Specifically, $\mathcal{T}(X)[a]$ is the percentage of the relevant items from X that belong to the topic a. We report *topdiv@k* averaged over all users.

5 RESULTS AND DISCUSSION

In this section we analyse both the effect of the evaluation protocol, and the merits of DPE for preference elicitation.

5.1 Comparison of Methodologies

Figure 1 shows our results on the Movielens 20M dataset. When using the standard TestItems approach, we see two effects.

Shallow vs. Deep Metrics. We see that at a shallow cut-off of 10, the Greedy baseline achieves the best performance both in terms of accuracy (graded and ungraded) and diversity. However, Valcarce et al. [55] found that deeper cut-offs are more reliable for offline evaluation, even when looking for optimal performance on lower cut-offs. We see that the relative merits change significantly when observing the same metrics at 100 cut-offs. First, in terms of graded relevance, the Greedy baseline is surpassed by DPE for most question counts. This effect is even more significant in the case of *P@100* and both diversity-sensitive metrics, *serendipity@100* and *topdiv@100*. As noted in section 4.2.2, this is the expected behavior of offline evaluation with the missing ratings effect.

TestItems vs. AvailablePreferences. Next, we analyze performance when using the *TestItems* versus the *AvailablePreferences* methodology. Recall that *AvailablePreferences* reduces the number of missing user preferences by adding the leftovers of the elicitation process as relevance judgments when computing metrics. We remind the reader that the comparison is fair between algorithms asking the same number of questions. When using this protocol, we see that the performance curves change dramatically. In particular, for every cut-off and metric, the Greedy baseline performs much worse than DPE. This pattern is not unexpected: Like many other elicitation methods, Greedy tries to maximize recommendation performance, and RSs are known to be biased to popularity. Therefore the Greedy approach is recreating the feedback-loop problem with each elicited item overspecializing the user profile to popular items. When evaluating on a small proportion of the relevance judgments, it performs well because popular items would also be more frequent in the test data. However, many relevant and non-popular items from the user preferences would not be recommended when using the

[1] https://www.coursera.org/specializations/recommender-systems, accessed May 2020

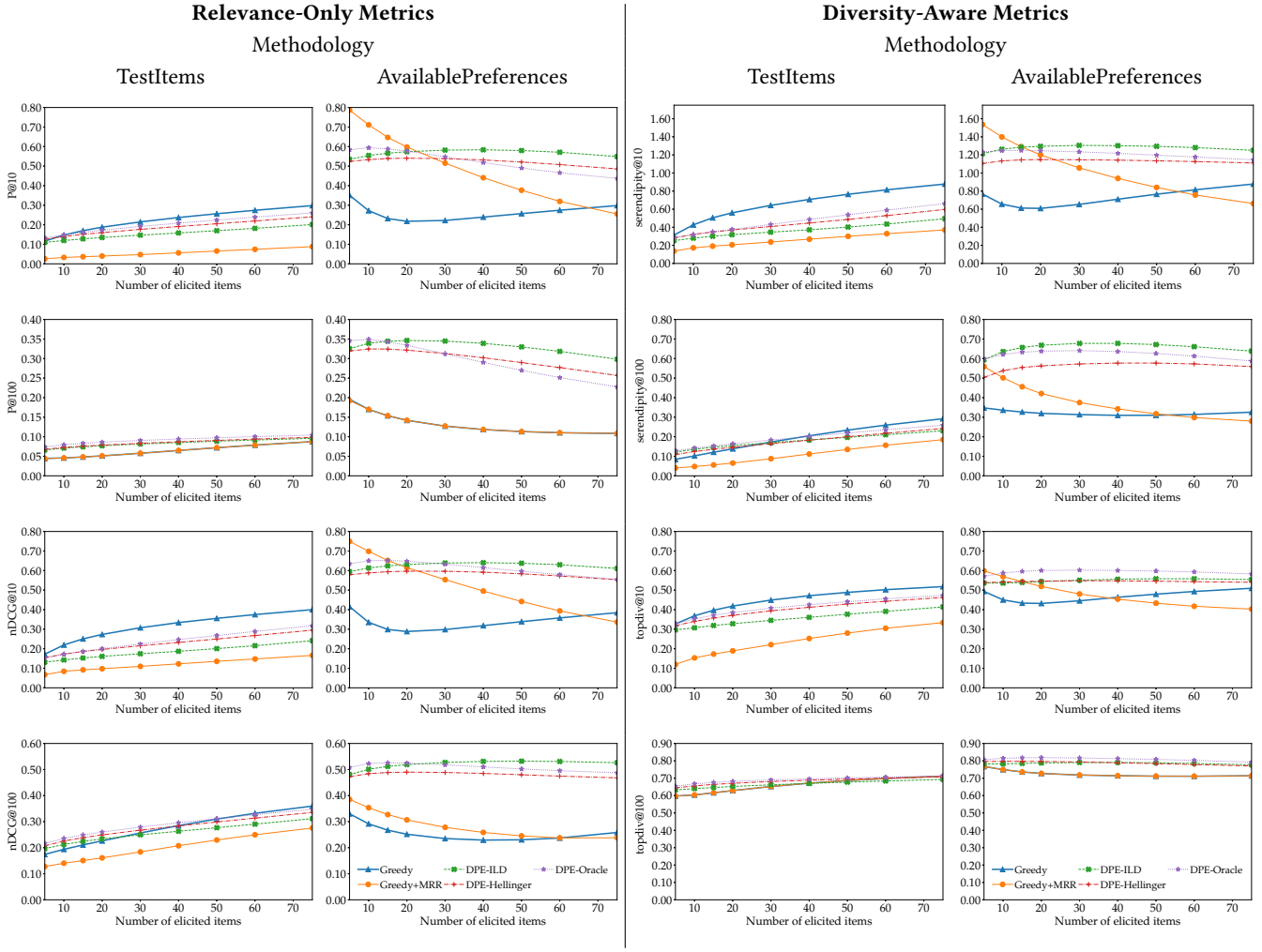

Figure 1: Results on the Movielens 20M dataset, comparing the TestItems and AvailablePreferences methodologies across eight different metrics.

Greedy baseline because it is unable to identify users' infrequent tastes. This apparent yet unreliable accuracy of popularity-biased recommendations has been identified before by Cañamares and Castells [8]. When using *AvailablePreferences*, we use the most information possible about the user's relevance preferences. Note that AvailablePreferences provides the same number of test items for all algorithms, but the leftovers from each elicitation algorithms will be different. It could be suspected that the leftovers from some models are easier to guess (i.e., more popular items) than from others. This *could* favor the recommendation performance of algorithms not eliciting those items. However, that possible bias can be easily excluded by observing the serendipity of the recommendations. As shown in the serendipity@k results, we see that DPE does not benefit from recommending popular items present in the test split. Rather the opposite is the case: it recommends less popular items substantially more often than baselines.

Moreover, we observe than when there are fewer missing preferences in the evaluation, the order of systems for the shallow cutoff is more similar to that at a deeper cutoff in the *TestItems* methodology. This finding confirms the need for the new methodology for preference elicitation and, at the same time, corroborates the suggestions of Valcarce et al. [55] of using deep cut-offs for having a more robust evaluation to mitigate the missing ratings effect.

There are another two important observations regarding the evaluation protocol. First, *topdiv* should only be considered at the 100 cut-off, because the number of categories in the collections is larger than 10: It is not possible to obtain a perfect score with a cutoff smaller than the number of categories. Second, when using *AvailablePreferences*, it is perhaps counter-intuitive that the metric values do not monotonically increase as more preferences are collected. This is caused by fewer "leftover" ratings in the test set as more preferences are elicited. In other words, the shape of the learning curve as the number of preference questions changes comes from the evaluation corpus not being fixed. Thus, how the value changes for a single algorithm is *not meaningful*. Rather, the comparison *between* different methods is a fair comparison, as for

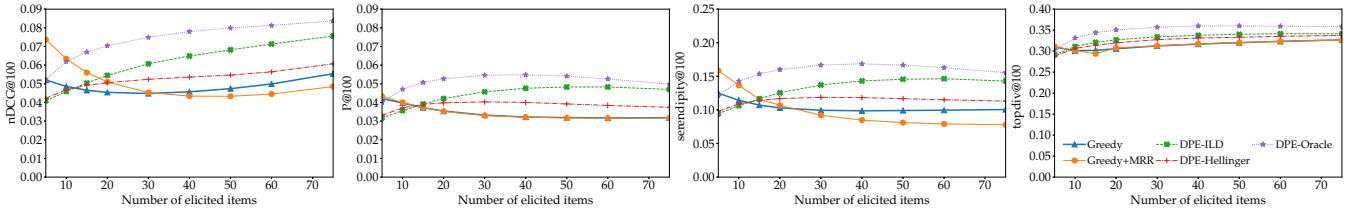

Figure 2: Results for the Amazon dataset for 100 cutoff under *AvailablePreferences* methodology.

each the number of preferences is used for computing the different metrics.

5.2 MABs for elicitation diversification

We analyze now the performance of DPE under the *AvailablePreferences* methodology on both the Movielens and Amazon datasets (Figures 1 and 2 respectively). When we consider the relevance-only metrics, we see that the DPE surpasses the Greedy baseline performance for all relevance metrics. Interestingly, we observe that the performance is quite stable with the number of elicited ratings, meaning that the ranking quality increases enough to compensate for the reduction of "leftover" relevance judgments, contrary to the Greedy method. Regarding the accuracy of the different arm ranker alternatives, the oracle-based strategy (DPE-Oracle) performs very well, as may be expected (because it always selects highly liked items). However, the broader exploration of the item-to-item diversification approaches of the other ranking alternatives produces a broader knowledge of the user preferences, resulting in better recommendations than with the oracle.

Regarding diversity, DPE-ILD is also the best alternative for the serendipity of the recommendations. Interestingly, DPE-Oracle outperforms the Hellinger method despite only applying topic level diversification on the Movielens dataset (Figure 1) and both Hellinger and ILD in the Amazon dataset (Figure 2). This shows the important role that thematic diversification plays in ranking diversification. The low serendipity values of the Greedy baseline show that it tends to rank relevant items more equally for every user. This result confirms the tendency of that approach to produce more recommendations biased towards popularity. On the other hand, concerning *topdiv*, DPE-Hellinger performs better than DPE-ILD when few items are elicited. In this case, as expected, the DPE-Oracle approach performs best, as it is applying only topic diversification over the most liked user items. Again the baselines achieve the worst values as they are not considering topical diversification.

Analyzing post-hoc diversification with MMR, we see an unbalanced behavior: MMR performs well with *few* elicited items, suggesting that when the recommendation algorithm lacks enough information, diversification increases the chance of stumbling upon user interests. However, at a certain point, the post-hoc diversification starts to introduce too many non-relevant items from the bottom of the ranking, quickly degrading accuracy. A similar pattern can be observed in the diversity metrics.

Finally, consider the differences among datasets. We can see that performance on the Amazon dataset is, as expected, much lower. Those numbers are due to the difficulty of the dataset given the high sparsity (fewer than four ratings per item on average). In general, however, the same trends hold. This is an important result: our proposal can improve the baselines figures even in such difficult situations where popularity should have a great advantage.

5.3 Summary

Our results show that our approach for tackling diversification of the *preference elicitation process* performs well. We see that diversifying the elicitation does not harm recommendation accuracy, but to the contrary improves accuracy (RQ1). We also see that a diversified elicited profile contributes to more unexpected and diverse recommendations (RQ2). Moreover, the results of our methods could be further improved by adjusting the exploration-exploitation trade-off of DPE by tuning the $k_a(0)$ parameters as we only used default initialization. These results also open the possibility to regulate diversification by selecting alternative arm ranking strategies to promote either item-to-item or topical diversity [47].

Finally, our experiments demonstrate how previous entropy reduction approaches to preference elicitation are favored by the sensitivity of the TestItems offline approach to the missing ratings problem. This problem is reduced by using AvailablePreferences, using the whole set of available user preferences for evaluation.

6 CONCLUSIONS

In this paper, we jointly addressed preference elicitation and diversification tasks. With this new approach, we obtain a broader representation of the user, resulting in better recommendations. Results show how a broader view of user interests results in important improvements over the state of the art methods not only in diversity, but also in terms of accuracy of the recommendations. We have also proposed a new evaluation methodology for preference elicitation to reduce the undesired effect of missing ratings, which tends to favor methods that over-weight popularity.

As future work, we will test alternative bandit configurations and arm ranking strategies. For instance, very frequently there are relationships among item categories that can be exploited, and may suit hierarchical MABs alternatives. Moreover, popular items may play a role in system validation, so the complete removal of popular items must be considered with caution. A greedy-like arm ranker strategy for combining both topical diversity and entropy reduction is worth exploring when more popularity is needed.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *Proc. RecSys.* ACM, 42–46.

[2] Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected. *ACM Trans.*

Intell. Syst. Technol. 5, 4 (Dec. 2014).
[3] Shipra Agrawal and Navin Goyal. 2013. Further optimal regret bounds for thompson sampling. In *Journal of Machine Learning Research*.
[4] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, Scrutable and Explainable User Models for Personalized Recommendation. In *Proceedings of SIGIR'19*. ACM, 265–274.
[5] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. 2011. Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison. In *Proceedings of RecSys '11*. ACM, 333–336.
[6] Keith Bradley and Barry Smyth. 2001. Improving Recommendation Diversity. *Business* (2001).
[7] Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci. 2015. User Personality and the New User Problem in a Context-Aware Point of Interest Recommender System. In *Information and Communication Technologies in Tourism 2015*. Springer, 537–549.
[8] Rocío Cañamares and Pablo Castells. 2018. Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. In *Proceedings of SIGIR '18*. ACM, 415–424.
[9] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR '98*. ACM, 335–336.
[10] Sylvain Castagnos, Armelle Brun, and Anne Boyer. 2013. When Diversity Is Needed... But Not Expected!. In *Proceedings of IMMM'13*. IARIA XPS Press, 44–50.
[11] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. *Novelty and Diversity in Recommender Systems*. Springer, 881–918.
[12] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. 2019. How Serendipity Improves User Satisfaction with Recommendations? A Large-Scale User Evaluation. In *Proceedings of WWW'19*. ACM, 240–250.
[13] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of KDD '16*. ACM, 815–824.
[14] Mehdi Elahi, Matthias Braunhofer, Tural Gurbanov, and Francesco Ricci. 2018. User Preference Elicitation, Rating Sparsity and Cold Start. In *Collaborative Recommendations*. WorldScientific, 253–294.
[15] Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *Manage. Sci.* 55, 5 (May 2009), 697–712.
[16] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In *Proceedings of RecSys '10*. ACM, 257–260.
[17] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. 2010. On Bootstrapping Recommender Systems. In *Proceedings of CIKM '10*. ACM, 1805–1808.
[18] Abhay S. Harpale and Yiming Yang. 2008. Personalized Active Learning for Collaborative Filtering. In *Proceedings of SIGIR '08*. ACM, 91–98.
[19] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (Dec. 2015).
[20] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of WWW '16*.
[21] E. Hellinger. 1909. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal fur die Reine und Angewandte Mathematik* (1909).
[22] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53.
[23] Abdollahpouri Himan, Mansoury Masoud, Burke Robin, and Mobasher Bamshad. 2019. The Unfairness of Popularity Bias in Recommendation. In *WS on Recommendation in Multi-stakeholder Environments – RecSys'19*. ACM.
[24] Junya Honda and Akimichi Takemura. 2014. Optimality of Thompson sampling for Gaussian bandits depends on priors. In *Artificial Intelligence and Statistics*. 375–383.
[25] Saikishore Kalloori, Francesco Ricci, and Rosella Gennari. 2018. Eliciting Pairwise Preferences in Recommender Systems. In *Proc. RecSys*. ACM, 329–337.
[26] Michael N. Katehakis and Arthur F. Veinott. 1987. The Multi-Armed Bandit Problem: Decomposition and Computation. *Math. Oper. Res.* 12, 2 (May 1987), 262–268.
[27] Jaya Kawale, Hung Bui, Branislav Kveton, Long Tran Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In *Proceedings of NIPS'15*. MIT Press, 1297–1305.
[28] Pushmeet Kohli, Mahyar Salek, and Greg Stoddard. 2013. A Fast Bandit Algorithm for Recommendations to Users with Heterogeneous Tastes. In *Proceedings of AAAI'13*. AAAI Press, 1135–1141.
[29] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A Survey of Serendipity in Recommender Systems. *Know.-Based Syst.* 111, C (Nov. 2016), 180–192.
[30] Dominik Kowald, Markus Schedl, and Elisabeth Lex. 2020. The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study. In *Proceedings of ECIR'2020*. Springer, 35–42.
[31] Matev Kunaver and Toma Porl. 2017. Diversity in Recommender Systems A Survey. *Know.-Based Syst.* 123, C (May 2017), 154–162.
[32] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *Proceedings of WWW '10*. ACM, 661–670.
[33] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative Filtering Bandits. In *Proceedings of SIGIR '16*. ACM, 539–548.
[34] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized News Recommendation Based on Click Behavior. In *Proceedings of IUI '10*. ACM, 31–40.
[35] David E. Losada, Javier Parapar, and Álvaro Barreiro. 2016. Feeling Lucky? Multi-Armed Bandits for Ordering Judgements in Pooling-Based Evaluation. In *Proceedings of SAC '16*. ACM, 1027–1034.
[36] David E. Losada, Javier Parapar, and Alvaro Barreiro. 2017. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing & Management* 53, 5 (2017), 1005 – 1025.
[37] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *Proceedings of CHI EA '06*. ACM, 1097–1101.
[38] Rouzbeh Meymandpour and Joseph G Davis. 2020. Measuring the diversity of recommendations: a preference-aware approach for evaluating and adjusting diversity. *Knowledge and Information Systems* 62, 2 (2020), 787–811.
[39] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. 2008. Metrics for Evaluating the Serendipity of Recommendation Lists. In *New Frontiers in Artificial Intelligence*. Springer, 40–46.
[40] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of WWW '14*. ACM, 677–686.
[41] Seung-Taek Park and Wei Chu. 2009. Pairwise Preference Regression for Cold-Start Recommendation. In *Proceedings of RecSys'09*. ACM, 21–28.
[42] Pearl Pu and Li Chen. 2008. User-Involved Preference Elicitation for Product Search and Recommender Systems. *AI Magazine* 29, 4 (2008).
[43] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning Diverse Rankings with Multi-Armed Bandits. In *Proceedings of ICML '08*. ACM, 784–791.
[44] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. 2002. Getting to Know You: Learning New User Preferences in Recommender Systems. In *Proceedings of IUI '02*. ACM, 127–134.
[45] Al Mamunur Rashid, George Karypis, and John Riedl. 2008. Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach. *SIGKDD Explor. Newsl.* 10, 2 (Dec. 2008), 90–100.
[46] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2010. *Recommender Systems Handbook* (1st ed.). Springer-Verlag.
[47] Hu Rong and Pu Pearl. 2011. Helping Users Perceive Recommendation Diversity. In *Proceedings of DiveRS 2011*. ACM, 6.
[48] Tim Salimans, Ulrich Paquet, and Thore Graepel. 2012. Collaborative Learning of Preference Rankings. In *Proceedings of RecSys '12*. ACM, 261–264.
[49] Javier Sanz-Cruzado, Pablo Castells, and Esther López. 2019. A Simple Multi-Armed Nearest-Neighbor Bandit for Interactive Recommendation. In *Proceedings of RecSys '19*. ACM, 358–362.
[50] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and Metrics for Cold-Start Recommendations. In *Proceedings of SIGIR '02*. ACM, 253–260.
[51] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference Elicitation as an Optimization Problem. In *Proc. RecSys*. ACM, 172–180.
[52] Guy Shani and Asela Gunawardana. 2011. *Evaluating Recommendation Systems*. Springer, 257–297.
[53] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit User Modeling for Personalized Search. In *Proceedings of CIKM '05*. ACM, 824–831.
[54] Yading Song, Simon Dixon, and Marcus Pearce. 2012. A survey of music recommendation systems and future perspectives. In *Proceedings of CMMR'12*, Vol. 4. 395–410.
[55] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2018. On the Robustness and Discriminative Power of Information Retrieval Metrics for Top-N Recommendation. In *Proceedings of RecSys '18*. ACM, 260–268.
[56] Daniel Valcarce, Javier Parapar, and Alvaro Barreiro. 2016. Language Models for Collaborative Filtering Neighbourhoods. In *Proceedings of ECIR' 16*, Vol. 9626. Springer, 614–625.
[57] Simon Wakeling, Paul Clough, Barbara Sen, and Lynn Silipigni Connaway. 2012. "Readers who borrowed this also borrowed…": recommender systems in UK libraries. *Library Hi Tech* 30, 1 (2012), 134–150.
[58] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H. Chi. 2015. Improving User Topic Interest Profiles by Behavior Factorization. In *Proceedings of WWW'15*. Republic and Canton of Geneva, Switzerland, 1406–1416.
[59] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of WWW '05*. ACM, 22–32.