Relevance meets Diversity: A User-Centric Framework for Knowledge Exploration through Recommendations

Anonymous Author(s)

ABSTRACT

Providing recommendations that are both *relevant* and *diverse* is a key consideration of modern recommender systems. Optimizing both of these measures presents a fundamental trade-off, as higher diversity typically comes at the cost of relevance, resulting in lower user engagement. Existing recommendation algorithms try to resolve this trade-off by combining the two measures, relevance and diversity, into one aim and then seeking recommendations that optimize the combined objective, for a given number of items to recommend. Traditional approaches, however, do not consider the user interaction with the recommended items.

In this paper, we put the *user* at the central stage, and build on the interplay between *relevance, diversity*, and *user behavior*. In contrast to applications where the goal is solely to maximize engagement, we focus on scenarios aiming at maximizing the total amount of knowledge encountered by the user. We use diversity as a surrogate of the amount of knowledge obtained by the user while interacting with the system, and we seek to maximize diversity. We propose a probabilistic user-behavior model in which users keep interacting with the recommender system as long as they receive relevant recommendations, but they may stop if the relevance of the recommended items drops. Thus, for a recommender system to achieve a high-diversity measure, it will need to produce recommendations that are *both relevant and diverse*.

Finally, we propose a novel recommendation strategy that combines relevance and diversity by a copula function. We conduct an extensive evaluation of the proposed methodology over multiple datasets, and we show that our strategy outperforms several state-of-the-art competitors. Our implementation is publicly available.[1]

ACM Reference Format:

Anonymous Author(s). 2024. Relevance meets Diversity: A User-Centric Framework for Knowledge Exploration through Recommendations. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems play a significant role in helping users discover new information and expand their knowledge base. Notable examples are the adoptions of recommendations for finding news articles or books to read [3, 54, 64], listening to enjoyable music [18, 45], visiting interesting locations [55, 58], and more. Recommender systems aim to predict and leverage users' interests to identify the portions of the catalog that match them, thus enabling efficient exploration of vast volumes of information and consequently offering benefits ranging from increased personalization and user satisfaction to improved engagement and resource efficiency.

Recommenders are primarily focused on maximizing relevance. However, from the standpoint of knowledge exploration, incorporating diversity into recommendations adds significant value, as emphasized in earlier research [24, 48]. Indeed, providing diverse recommendations can be critical in mitigating detrimental consequences, such as being trapped in *rabbit holes* in platforms like Youtube [20, 35, 42, 51] or Reddit [40], where the algorithm may lead the user to consume limited types of content.

To achieve a balance between relevance and diversity, current methods merge these two metrics into a single objective for optimization. However, they overlook user behavior and how users interact with the recommended list of items. For instance, typical approaches assume a fixed number of interactions between the user and the algorithm, disregarding any reactions or refusals from the user during the exploration process. Indeed, users might reject recommended items and quit the process.

In this paper we propose a new framework for recommender systems, where we place the user at the forefront. We consider the interaction of the user with the algorithm to be a *knowledge-exploration task*, where recommendations enable exploration. The interaction of the user with the system is guided via a *user-behavior model*, i.e., the propensity of a user to accept or reject recommendations according to their preferences and patience. As the objective is to maximize the amount of knowledge that a user acquires during exploration, we model the knowledge accrued by the user using a *diversity* measure, which we consequently aim at maximizing.

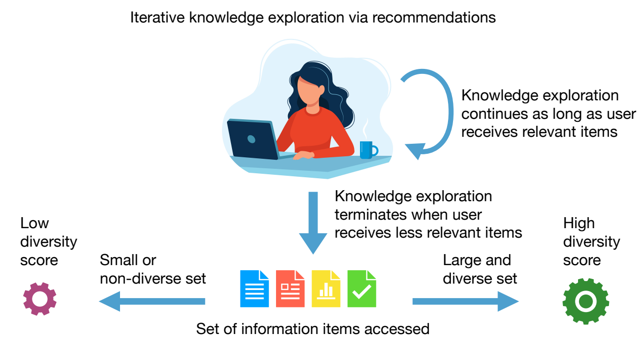


Figure 1: The knowledge-exploration process, illustrating the interplay among *relevance, diversity,* and *user behavior.*

[1] https://anonymous.4open.science/r/EXPLORE-AA84/

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

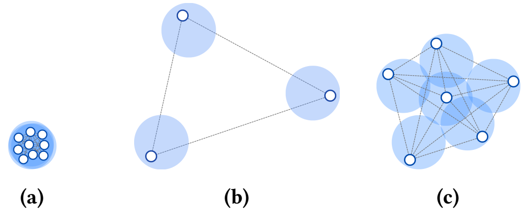


Figure 2: Illustration of the impact of different recommendation strategies. White points are recommended items, blue circles indicate information coverage. (a) High relevance, low diversity (e.g., all about 'technology'); (b) High diversity, likely non-relevant (e.g., 'technology,' 'religion,' 'lifestyle'); (c) Optimal balance: Relevant and diverse, keeping user engaged (e.g., 'technology,' 'science,' 'engineering.').

Notably, although diversity is the sole optimization objective, the coupling of the exploration task with the user-behavior model implies that the recommendation system is required to produce recommendations that are *both relevant and diverse*.

We illustrate the proposed concept of *"knowledge exploration via recommendations"* with the following example.

Example. Alice interacts with a news recommender system for finding interesting news articles to read. The knowledge-exploration process is iterative, and is depicted in Figure 1. At each step, the system recommends a set of news articles to Alice, and Alice clicks on some article to read. At some point, Alice can decide to quit, either because she received enough information, or because the recommendations are not very interesting to her, or simply because she got bored. Our goal is to design a recommender system that maximizes the amount of knowledge received by Alice. The challenge is to strike a balance between diversity and relevance to keep Alice engaged while exploring interesting topics, avoiding scenarios where recommendations are either too focused (Figure 2(a)) or too diverse and irrelevant (Figure 2(b)). Our aim is to create an ideal scenario (Figure 2(c)) where Alice explores many relevant yet diverse topics, enriching her knowledge.

Motivated by the previous example, we propose a novel framework where relevance governs the termination of exploration, while the overall quality is measured by diversity. We instantiate our model using two standard notions of diversity, one based on coverage and the other based on pair-wise distances [4, 9, 13]. Both diversity notions, coverage and pairwise distances, can be defined using an underlying space of user-to-item ratings or categories/topics.

Finally, we propose a novel recommendation strategy that combines relevance and diversity by a copula function. We perform an extensive evaluation of the proposed framework and strategy using five benchmark datasets publicly available, and show that our strategy outperforms several state-of-the-art competitors.

Our contributions are summarized as follows:

- We develop a user-centric model for knowledge exploration via recommendations; our framework takes into consideration the interplay among relevance, diversity, and user behavior.
- We instantiate our model with two diversity measures, defined over user-to-item ratings or categories/topics.
- We propose a recommendation strategy that accounts for both diversity and relevance when providing suggestions.
- We conduct an extensive analysis over multiple benchmark datasets and several competitors to show the effectiveness of our proposal in the suggested framework.

The rest of the paper is structured as follows. Section 2 presents the related work in terms of user modeling and diversity in recommendations. Section 3 presents our problem definition and methodology. In Section 4 we present our recommendation strategy. Experimental results are reported in Section 5, and finally Section 6 concludes the paper and provides pointers for future extensions.

2 RELATED WORK

User modeling in recommender systems. The effects of user behavior in recommender systems, in terms of novelty and diversity, have gained a lot of attention in recent years. Analysis can be conducted by either running user studies [28, 29, 33, 62], or by means of simulation [23, 50, 57]. Analyzing the choices made by actual users can yield more dependable outcomes; however, it also requires creating an effective recommendation system and engaging users for conducting comprehensive studies.

On the other hand, simulating user choices is a more straightforward method, allowing for testing several system configurations at no expense. However, it requires a realistic model of *user behavior*. To address this challenge, several user-behavior models have been proposed in the literature. Hazrati and Ricci [22] model the probability that a user picks a recommended item as being proportional to the utility of the item. Similarly, Bountouridis et al. [7] propose a simulation framework in which users decide to interact with a certain number of items per iteration, according to their given preferences. Szlávik et al. [50] present three different user-behavior models, by imposing that users either blindly follow recommendations and choose the most popular items, or completely ignore suggestions and pick items randomly.

The aforementioned models present certain limitations, namely users necessarily have to pick an item, i.e., they cannot leave the application, and second, the selection probability stays constant over time. We overcome these limitations by modeling a *quitting probability*, according to which users can interrupt their interaction with the recommender system. We assume that the quitting probability depends on the utility of the recommended items and on the user *patience*, which degrades over time.

Notably, with our framework, we leverage the intrinsic interplay among relevance, diversity, and user behavior, since successful recommendation strategies need to ensure that they provide recommendations that are both relevant and diverse.

Diversity in recommendation. Diversity in recommendations has been acknowledged as a crucial issue [10, 24, 48, 63], and over the past decade, it has received considerable attention [1, 2, 11, 25, 52]. Several online and targeted user studies assessed the increase in user satisfaction when diversity is incorporated into the list of suggested items [10, 25]. For example, Allison et al. [12] show that, if diversity (besides other objectives) is not taken into account, the interactions between users and recommender systems are prone to homogenization and, consequently, low utility.

The challenge of striking a balance between diversity and relevance has been explored both in the context of recommender systems and in the broader domain of information retrieval. For instance, one of the most popular methods in the literature of information retrieval is the *maximal marginal relevance* (MMR) [9]. It employs a weighted linear combination of scores that evaluate both utility and diversity, offering a systematic way to address this critical aspect. In the specific context of recommender systems, Ziegler et al. [63] introduced one of the earliest methods for enhancing diversity. They use a greedy selection approach, where they pick items that minimize the similarity within a recommended list. Liu et al. [32] present a solution based on random walks for the so-called *accuracy-diversity dilemma*, i.e., the challenge in finding a profitable trade-off between the two measures. This concept is also known as *calibration*, as mentioned by Steck [49], and refers to the algorithm's capability to produce suggestions that do not under-represent (or ignore) the user's secondary areas of interest.

Several re-ranking strategies have also been proposed: Ashkan et al. [4] propose to greedily select items by maximizing the utility of a submodular function; Sha et al. [47] suggest to optimize the diversity loss of items using probabilistic matrix factorization; Chen et al. [13] propose a determinantal point process (DPP) to re-rank the recommended items so as to maximize the determinant on the items' similarity matrix. Hansen et al. [19] investigate the impact of diversity on music consumption, and propose two innovative models: a feed-forward neural ranker that produces dynamic user embedding, and a reinforcement learning-based ranker optimized on the track relevance. *Reinforcement learning* is indeed a suitable solution for addressing the diversity problem. It plays a role in the work by Parapar and Radlinski [37], where diversity is induced by adopting multi-armed bandits in the elicitation phase; and in the online learning framework proposed by Yue and Guestrin [59], where diversification is obtained by carefully balancing the exploration and exploitation of users' preferences and interests. Notably, these reinforcement learning-based approaches typically require a lengthy training phase, which can often be prone to stability issues.

Several other *neural-network models* have been applied to address the diversity problem. Gao et al. [16] adopt a *variational autoencoder* to induce targeted (i.e., topical) diversity. Liang et al. [30] propose a *bilateral branch network* to achieve a good trade-off between relevance and diversity, defined at either domain or user level. Zheng et al. [61] present a *graph neural network* for diversified recommendations, where node neighbors are selected based on inverse category frequency, together with negative sampling for inducing diverse items in the embedding space.

In contrast to most of the approaches mentioned earlier, the recommendation strategy we introduce, EXPLORE, does not necessitate any form of training or hyperparameter tuning, it is computationally efficient, and is shown to provide both highly relevant and diverse suggestions.

3 USER MODEL AND PROBLEM FORMULATION

We consider a typical recommendation setting in which we have a set of m users $\mathcal{U}$ and a set of n items $\mathcal{I}$. We also consider a function $\mathcal{R} : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$ that provides us with a relevance score $\mathcal{R}(u, i)$, for

Algorithm 1 Simulation process for user u

Input: $u, \mathcal{I}, \mathcal{S}, \mathcal{R}$
Output: $\mathcal{X}$

$\mathcal{X} \leftarrow \emptyset$
quit $\leftarrow$ False
while not *quit* **do**
  $\mathbf{L}_t = [i_i, i_2, ..., i_k] \leftarrow \mathcal{S}(\mathcal{R}(u, \mathcal{I} \setminus \mathcal{X}), \mathcal{X})$
  examining $\mathbf{L}_t \leftarrow$ Algorithm 2
  if u does not quit **then**
    $i \leftarrow$ picked item
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{i\}$
  else
    quit $\leftarrow$ True
  end if
end while

Algorithm 2 User behavior at step t

Input: $\mathbf{L}_t$
Output: $i \in \mathbf{L}_t$ or *quits*

interest $\leftarrow$ False
for $j = 1, \ldots, k$ **do**
  $i \leftarrow \mathbf{L}_t[j]$
  quitting $\leftarrow$ with probability η_t
  if u quits **then**
    return
  else
    examining $i \leftarrow$ with probability q_i
    if i is interesting **then**
      interest $\leftarrow$ True
    end if
  end if
end for
if not *interest* **then**
  return
end if
for $j = 1, \ldots, k$ **do**
  $i \leftarrow \mathbf{L}_t[j]$
  consuming $i \leftarrow$ with probability p_i
  if u consumes i **then**
    return i
  end if
end for

each user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$. We assume that the function $\mathcal{R}$ can be computed by a black-box method, and state-of-the-art relevance-scoring functions can be employed, such as content similarity [38], collaborative filtering [44], or a combination of both [8]. Our goal in this paper is to create lists of diverse recommendations using as a black box such relevance-scoring functions, rather than devising a new relevance-scoring function $\mathcal{R}$.

Item-to-item distance function. We next discuss how to define a distance function between pairs of items in $\mathcal{I}$, which will be used in one of our two diversity definitions.

Given an item $i \in \mathcal{I}$, we denote by $\mathbf{x}_i$ the vector of *users* with

$$x_{iu} = \begin{cases} 1, & \text{if user } u \text{ interacted with item } i, \\ 0, & \text{otherwise.} \end{cases}$$

The vectors $\{\mathbf{x}_i\}$ can be retrieved by user-log data. A more fine-grained representation of vectors $\{\mathbf{x}_i\}$ beyond binary is also possible, for instance, using numerical values that represent the *rating* of user u for item i, if such information is available.

An alternative approach is to use categories (or keywords, or genres, depending on the application). In particular, we consider a set of categories C, and we define $\mathbf{y}_i$ to be a *category* vector, for item $i \in \mathcal{I}$, where

$$y_{ic} = \begin{cases} 1, & \text{if category } c \text{ relates to item } i, \\ 0, & \text{otherwise.} \end{cases}$$

Given two items $i, j \in \mathcal{I}$, we hence define their *distance* as the *weighted Jaccard distance*

$$d(i,j) = 1 - \frac{\sum_{w \in \mathcal{W}} \min\{z_{iw}, z_{jw}\}}{\sum_{w \in \mathcal{W}} \max\{z_{iw}, z_{jw}\}}, \tag{1}$$

where $\mathcal{W}$ is either the set of users $\mathcal{U}$ or the set of categories C, and accordingly, $\mathbf{z}_i$ is the *user vector* or the *category vector* of item i.

Finally, we note that other state-of-the-art distance functions can also be used, such as Euclidean distance, cosine similarity, or Minkowski distance [6]. We do not investigate what is the best distance function to be used, as this is orthogonal to our study and beyond the scope of this paper.

Diversity. Given a set of items $\mathcal{X} \subseteq \mathcal{I}$, we define the *diversity* of the set $\mathcal{X}$. We explore two different definitions of diversity.

Our first definition is based on the concept of *coverage*. It assesses the degree to which the items within $\mathcal{X}$ adequately represent the entire range of categories C. In particular, for a set of items $\mathcal{X} \subseteq \mathcal{I}$, we define its *coverage-based diversity* as

$$\mathrm{div}_C(\mathcal{X}) = \frac{1}{|C|} \left\| \bigvee_{i \in \mathcal{X}} \mathbf{y}_i \right\|_0, \tag{2}$$

where $\|\cdot\|_0$ returns the number of non-zero entries of the binary vector $\bigvee_{i \in \mathcal{X}} \mathbf{y}_i$. Notice that the metric div_C is scaled to fall within the range of 0 to 1, considering the total number of categories in C. Also, it is worth highlighting that div_C tends to favour larger $\mathcal{X}$ sizes, as they typically cover a wider range of categories. Additionally, div_C naturally prefers items that individually provide extensive coverage.

Our second measure of diversity employs the distance function d that we defined in the previous paragraph. In particular, for a set of items $\mathcal{X} \subseteq \mathcal{I}$ with $|\mathcal{X}| \geq 2$, we define its *distance-based diversity* as

$$\mathrm{div}_D(\mathcal{X}) = \frac{1}{|\mathcal{X}|-1} \sum_{i \in \mathcal{X}} \sum_{j \in \mathcal{X}} d(i,j), \tag{3}$$

and we define $\mathrm{div}_D(\mathcal{X}) = 0$, if $|\mathcal{X}| < 2$. Notice that the number of terms in div_D is quadratic with respect to $|\mathcal{X}|$. By normalizing with $(|\mathcal{X}|-1)$ the dependence becomes linear in $|\mathcal{X}|$. As with div_C, the div_D metric favors larger sets, in addition to favoring items whose distance is large to each other.

User model. A central aspect of our approach is that we aim to evaluate the quality of a recommendation algorithm $\mathcal{S}$ in the context of the user response to items recommended by $\mathcal{S}$. We view the user-algorithm interaction as a dynamic knowledge-exploration process, in which the algorithm recommends items to the user, and the user interacts with the recommended items. The knowledge-exploration process continues as long as the recommended items are of interest to the user. If the recommended items are not interesting enough (meaning, if they have low relevance for the user) the user may (stochastically) terminate the exploration process.

To formalize the exploration process between the user and the recommendation algorithm $\mathcal{S}$, which is needed to evaluate the quality of $\mathcal{S}$, we propose a *user model*. Our model is specified in terms of a relevance-scoring function $\mathcal{R}$, which guides the behavior of the user, and in terms of a recommendation algorithm $\mathcal{S}$, which enacts the choices within $\mathcal{S}$.

Our user model, which formalizes knowledge-exploration as an iterative process, is described as follows.

(1) The set of items that the user interacts with during the exploration process is denoted by $\mathcal{X}$. Initially $\mathcal{X}$ is empty.
(2) In the t-th step, the recommendation algorithm $\mathcal{S}$ generates a list of items $\mathbf{L}_t$ to present to the user. The user examines these items in a specified order.
(3) At any point in the current step, the user has the option to quit. The likelihood of quitting (to be quantified later) depends on two factors: the relevance of the recommended items and the user's patience. If the user fails to find interesting items in list $\mathbf{L}_t$ or if they stochastically run out of patience, they may opt to conclude the exploration process.
(4) If the user does not quit, with a certain probability that depends on the relevance of the recommended items (and which we quantify later), they select an item i from the list $\mathbf{L}_t$ and interact with it. The item i is added to the set $\mathcal{X}$ and the exploration process continues.
(5) Upon quitting, the total score achieved by the recommendation algorithm $\mathcal{S}$ is determined to be $\mathrm{div}(\mathcal{X})$, where div is one of our diversity functions, div_C or div_D. This score reflects the diversity in the items the user has interacted with throughout the exploration process. We denote the final number of steps performed by the user as κ.

Algorithm 1 depicts the pseudo-code of the overall exploration process.

To fully specify the user model we need to describe in more detail the probability that the user selects an item to interact with, as well as the probability of quitting the exploration. Before presenting more details about these aspects of the model, we first formalize the problem of designing a recommendation algorithm in the context of our user model.

The recommendation task (problem statement). The algorithmic problem that we address in this paper is the following.

Problem 1. *Given a set of items $\mathcal{I}$, a set of users $\mathcal{U}$, a relevance-scoring function $\mathcal{R} : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$, a diversity function $\mathrm{div} : 2^{\mathcal{I}} \to \mathbb{R}$, and a user model for knowledge-exploration as the one described in the previous paragraph, the goal is to design a recommendation algorithm $\mathcal{S}$ that maximizes the diversity score $\mathrm{div}(\mathcal{X})$ for the set of items $\mathcal{X}$ that a user $u \in \mathcal{U}$ interacts with.*

Item selection. We now discuss step (4) of the iterative knowledge-exploration user model presented in the previous paragraph, that is, we specify how we model the probability that a user selects an item i from the list $\mathbf{L}_t$ to interact with. We first assume that a user does not quit exploration, i.e., that they have enough patience to explore the whole $\mathbf{L}_t$ and that they find at least a relevant item within it (see next paragraph). In that case, the user selects an item i from $\mathbf{L}_t$ with probability proportional to the relevance of i for that user u, that is, $p_i = \frac{\mathcal{R}(u,i)}{\sum_{j \in \mathbf{L}_t} \mathcal{R}(u,j)}$. As noted before, the selected item i is added to the set of interacted items $\mathcal{X}$.

Quitting exploration. Last, we discuss step (3) in our user model, that is, how we model the probability that a user quits the exploration process. A sensible model for the quitting probability is crucial in our knowledge-exploration model, since we want to mimic user behavior as realistically as possible. In particular, we take into consideration two aspects: (*i*) users decide to interact with the recommended items according to their relevance; and (*ii*) users' desire for exploration degrades with time, i.e., users get bored.

In the model we propose, a user examines the items in the list $\mathbf{L}_t$ sequentially. Upon examining an item $i \in \mathbf{L}_t$ the user decides with probability η_t to quit exploration due to worn out at step t. We refer to this as the *weariness* probability. The weariness probability η_t, which is discussed in more detail below, models the user's decline of interest in exploration as a function of time, and it depends on the current step t in the exploration process.

If the user does not quit, they decide whether item i is interesting to explore. The latter is decided again stochastically with Bernoullian probability q_i, which is a function of the relevance score $\mathcal{R}(u,i)$.[2] Thus, the probability q_i models the user's interest in an item according to its relevance. The examination of the list $\mathbf{L}_t$ continues until the user decides to quit or decides that there is at least one item that is interesting to explore. Thus, the probability that the user quits examining the list $\mathbf{L}_t$ without identifying any item to explore is

$$\begin{aligned} Q_t &= \{\text{pr. quitting after the first item}\} + \ldots + \\ &\quad \{\text{pr. quitting after the last item}\} \\ &= \sum_{j=1}^{|\mathbf{L}_t|} \eta_t (1-\eta_t)^{j-1} \prod_{i=1}^{j-1} (1-q_i). \end{aligned} \tag{4}$$

The last ingredient in our model is to quantify the weariness probability η_t at step t. This probability models the user's increasing impatience or boredom as their interaction continues. To achieve this, we employ the Weibull distribution [36], which has been previously used to model web page dwell times and session lengths in web page navigation [31].

The Weibull distribution is described by two parameters, λ and γ, where $\lambda > 0$ is the scale parameter and $\gamma > 0$ is the shape parameter of the distribution. In particular, we set the weariness probability η_t by resorting to the discrete version of the Weibull Distribution [43]:

$$\eta_t = 1 - q^{(t+1)^\gamma - t^\gamma}, \tag{5}$$

where $q = e^{-1/\lambda^\gamma}$, $0 \le q \le 1$.

The shape parameter γ controls the "aging" of the process. For $\gamma = 1$ the weariness probability remains constant, and the resulting distribution becomes an exponential distribution, while for $\gamma > 1$, the weariness probability increases over time — modeling the tiredness of the user.[3]

We can use the analytical properties of the Weibull distribution to obtain the expected number of steps in the exploration process, for the case that all recommended items are maximally relevant, i.e., $q_i = 1$ for all $i \in \mathbf{L}_t$. In this case, there will be exactly one coin-flip for quitting exploration for each list $\mathbf{L}_t$, and thus, $Q_t = \eta_t$, for all t. The overall quitting probability Q_T is then

$$\begin{aligned} Q_T &= \{\text{pr. quitting at step } 1\} + \ldots + \\ &\quad \{\text{pr. quitting at step } t\} + \ldots \\ &= \sum_{t=1}^{\infty} Q_t \prod_{j=0}^{t-1} (1-Q_j) \\ &= \sum_{t=1}^{\infty} \left(1 - q^{(t+1)^\gamma - t^\gamma}\right) \prod_{j=0}^{t-1} q^{(j+1)^\gamma - j^\gamma} \\ &= \sum_{t=1}^{\infty} \left(1 - q^{(t+1)^\gamma - t^\gamma}\right) q^{t^\gamma} \\ &= \sum_{t=1}^{\infty} q^{t^\gamma} - q^{(t+1)^\gamma}. \end{aligned} \tag{6}$$

The expected number of steps $\mathbb{E}[\text{steps}]$ examined by a user before quitting (or equivalently, the number of items in $\mathcal{X}$) is hence given by

$$\mathbb{E}[\text{steps}] = \sum_{t=1}^{\infty} t \left(q^{t^\gamma} - q^{(t+1)^\gamma}\right). \tag{7}$$

Although lacking closed-form analytical expressions, Khan et al. [26] show that it is bounded by the expectation $\mu = \lambda\,\Gamma(1+1/\gamma)$ of the Weibull distribution in the continuous setting [36] as

$$\mu < \mathbb{E}[\text{steps}] < \mu + 1, \tag{8}$$

which provides an algebraic relationship between the λ parameter of the Weibull distribution and the admissible range for the expected number of steps.

Note that, if the relevance of the recommended items is less than 1, it is possible to get more than one coin-flip for quitting exploration in each list $\mathbf{L}_t$. In this case, the right-hand side of Equation (7) provides an upper bound on the expected number of steps during exploration.

A notation table can be found in the Appendix (Table 7).

Remarks on the proposed model. We observe that, as intended, our model captures both the relevance of the recommended items and the natural tiredness of users with exploration over time. For fixed values of the Weibull distribution parameters λ and γ, which control scaling and aging, the users' time for exploration increases with the relevance of the recommended items. Furthermore, the *ordering* of the items in the list $\mathbf{L}_t$ is important, and thus, we are viewing the recommendation list as a sequence, and not just as a set. This aspect would have implications on how to pick the appropriate recommendation strategy, but also on the objective (diversity) function, since it can affect the choices of the user.

4 RECOMMENDATION STRATEGY

In this section, we present our recommendation strategy for the proposed knowledge-exploration framework. Recall that the recommendation task is displayed as Problem 1.

[2] In our experiments, q_i is obtained by normalizing $\mathcal{R}(u,i)$ into the $[0,1]$ interval by considering the maximum relevance range.

[3] For $\gamma < 1$ the weariness probability decreases over time.

The core of the problem is to construct a list of recommendations $\mathbf{L}_t$ of size $\|\mathbf{L}_t\| = k$ for the t-th step of exploration, for a given user $u \in \mathcal{U}$. We assume that $\mathcal{X}_t$ is the set of items that the user has interacted with at step t, where $\mathcal{X}_1 = \emptyset$. We define $\mathcal{J}_t = \mathcal{I} \setminus \mathcal{X}_t$ to be set of items that are available for recommendation, that is, all items except the ones that the user has already interacted with.

For a user u and each item in the candidate set $i \in \mathcal{J}_t$ we consider its relevance score $\mathcal{R}_i = \mathcal{R}(u, i)$ and its *marginal diversity*

$$\mathcal{T}_i = \operatorname{div}(\mathcal{X}_t \cup \{i\}) - \operatorname{div}(\mathcal{X}_t), \tag{9}$$

with respect to the interaction set $\mathcal{X}_t$, where $\operatorname{div} \in \{\operatorname{div}_D, \operatorname{div}_C\}$. We denote $\mathcal{T}_i = \mathcal{D}_i$ when the distance diversity function div_D is used, and $\mathcal{T}_i = C_i$ when the coverage diversity function div_C is used. Intuitively, $\mathcal{D}_i$ represents the distance of i from all the items in the interaction set $\mathcal{X}_t$, while C_i represents the additional coverage that i provides.[4] Given $\mathcal{P}_i \in \{\mathcal{R}_i, \mathcal{T}_i\}$, we also denote the min-max normalization of the score $\mathcal{P}$ as $\widehat{\mathcal{P}}_i = (\mathcal{P}_i - \mathcal{P}_{min})/(\mathcal{P}_{max} - \mathcal{P}_{min})$, where $\mathcal{P}_{max}$ and $\mathcal{P}_{min}$ are the maximum and minimum values of $\mathcal{P}$, respectively, over all items in $\mathcal{X}_t$.

Our strategy for constructing the recommendation list $\mathbf{L}_t$ is to combine relevance and diversity into one score. For each item i with relevance $\mathcal{R}_i$ and diversity $\mathcal{T}_i$, we compute the combined score $\mathcal{Z}_i$ by adopting the Clayton copula function [14]

$$\mathcal{Z}_i = \left[\widehat{\mathcal{R}}_i^{-\alpha} + \widehat{\mathcal{T}}_i^{-\alpha} - 1\right]^{-1/\alpha}, \tag{10}$$

where $\alpha > 0$ is a regularization parameter. The list $\mathbf{L}_t$ is then formed by selecting the top-k items from $\mathcal{J}_t$ according to their combined score $\mathcal{Z}_i$.

We refer to this strategy as EXPLORE. When the distance diversity function is used we refer to it as EXPLORE-*D*, and when coverage diversity is used we refer to it as EXPLORE-*C*. A final word on the justification of using the copula function (10). Copulas are functions able to model the cumulative joint distribution of uniform marginal distributions. In general, they are used to represent correlation and dependencies of high-dimensional random variables [34, 39, 53, 60]. The Clayton copula function approaches 1 when both the input variables u, v are maximized, and it is minimized when either of them is 0. The α parameter governs the steepness and folding of the surface: the higher the value of α, the more stooped the function is when $u = v$.

The complexity of the algorithm is discussed in the Appendix.

5 EXPERIMENTS

In this section, we assess the performance of our recommendation strategy, either EXPLORE-*D* or EXPLORE-*C*, in balancing accuracy and diversity. We also compare its effectiveness with several state-of-the-art competitors within the proposed knowledge-exploration framework.

5.1 Datasets

We use five benchmark datasets, freely available online. We ensure that all datasets have category information, which is used by our diversity measures.

[4]At the beginning of the exploration process (when $\mathcal{X}_t = \emptyset$), if $\mathcal{T}_i = \mathcal{D}_i$, the strategy samples a highly relevant item i_r so that $\mathcal{D}_i = d(i, i_r)$; if $\mathcal{T}_i = C_i$, then $C_i = \mathbf{y}_i$, thus picking the item that individually provides the highest coverage.

Table 1: Dataset statistics and mean Jaccard distances with respect to users ($\hat{D}_U$) and categories ($\hat{D}_C$).

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	#Ratings	$\hat{D}_U$	$\hat{D}_C$
Movielens-1M	6 040	3 706	1 000 208	0.97	**0.83**
Coat	290	300	6 960	0.97	**0.73**
KuaiRec-2.0	1 411	3 327	4 676 570	**0.35**	0.91
Netflix-Prize	4 999	1 112	557 176	0.95	**0.83**
Yahoo-R2	21 181	3 000	963 296	0.99	**0.26**

Movielens-1M [21]:[5] A popular dataset with movie ratings in the range $[1, 5]$, and movie genres.

Coat [46]:[6] Ratings on coats in the range $[1, 5]$, and information on coats' properties.

KuaiRec-2.0 [15]:[7] A recommendation log from a video-sharing mobile app. Context information is provided, such as *play duration*, *video duration*, and *watch ratio*. We convert the watch ratios into ratings by interpolating the values from $[0, 2]$ to $[1, 5]$, where 0 represents "never watched" and 2 represents "watched twice." We use the *small* version of the dataset.

Netflix-Prize [5]:[8] Movie ratings in the range $[1, 5]$. We adopt a smaller sample of the original dataset by randomly selecting 5 000 items and discard the users with less than 20 interactions. Movie categories are acquired from a dataset using the IMDB database.[9].

Yahoo-R2:[10] Song ratings in the range $[1, 5]$. Each item is accompanied by artist, album, and genre information. We randomly sample 3 000 items and discard users with less than 20 interactions.

Table 1 provides a summary of the dataset properties, which include the number of users ($|\mathcal{U}|$), the number of items ($|\mathcal{I}|$), the number of ratings (#Ratings), and the distribution of item distances, calculated based on either users or categories. During our experiments, we use Equation (1) with the distance that exhibits the lowest mean for each dataset. This approach helps us avoid potential bias from large distance values, which could otherwise hinder the effectiveness of the approaches.

5.2 Competing recommendation strategies

We evaluate our recommendation algorithm, EXPLORE, against the following baseline and state-of-art strategies that have been designed for the task of increasing diversity in recommender systems.

Relevance: This approach recommends the k most relevant items, making it a fundamental baseline. Since this strategy is solely focused on maximizing relevance, it represents the most straightforward and basic diversity method, and any other approach must outperform it to be deemed effective.

Maximal marginal relevance (MMR) [9]: A classic method used to balance relevance and diversity, performed by optimizing the following marginal relevance:

$$\text{MMR} = \operatorname*{argmax}_{i \notin L} \left\{ \beta \mathcal{R}(u, i) - (1 - \beta) \max_{j \in L} \mathbf{S}_{i,j} \right\},$$

[5]https://grouplens.org/datasets/movielens/1m/
[6]https://www.cs.cornell.edu/~schnabts/mnar/
[7]https://kuairec.com/
[8]https://www.kaggle.com/datasets/rishitjavia/netflix-movie-rating-dataset
[9]https://github.com/tommasocarraro/netflix-prize-with-genres
[10]https://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67

where $\mathbf{S}_{i,j} = 1 - d(i, j)$. In our experiments, we set $\beta = 0.5$.

DUM [4]: This strategy aims at diversifying the suggestions by performing the following diversity-weighted utility maximization:

$$\text{DUM} = \underset{L \in \Pi}{\text{argmax}} \sum_{h=1}^{k} \left[f\left(L_{[:h]}\right) - f\left(L_{[:h-1]}\right) \right] \mathcal{R}(u, i_h),$$

where Π denotes all possible permutations of L, $L_{[:h]}$ represents the list up to the h-th element i_h, and $f(X) = \sum_{c \in C} \mathbb{1}\{\text{exists } i \in X : i \text{ covers category } c\}$ is the number of categories in X. Hence, the function maximizes the relevance of the recommended items weighted by the increase in their coverage.

DPP [13]: This method utilizes *determinantal point processes* and maximizes diversity by iteratively selecting the item i that maximizes the determinant of the item-item similarity matrix $\mathbf{S}$ defined on a subset of items:

$$\text{DPP} = \underset{i \notin L}{\text{argmax}} \left\{ \log\det(\mathbf{S}_{L \cup \{i\}}) - \log\det(\mathbf{S}_L) \right\}.$$

DGREC [56]: A *graph neural network* (GNN) based recommender that aims at finding a subset of diverse neighbors as well as maximizing the coverage of categories, by optimizing the loss function:

$$\mathcal{L}_{\text{DGREC}} = \sum_{(u,i) \in E} w_{\mathbf{y}_i} \mathcal{L}_{\text{BPR}}(u, i, j) + \lambda ||\Theta||_2^2,$$

where $w_{\mathbf{y}_i}$ is the weight for each sample based on its category, λ is a regularization factor, and $\mathcal{L}_{\text{BPR}}$ is the Bayesian personalized ranking loss [41] .

Notably, these competitors exhibit significant heterogeneity both in terms of the approaches they employ as well as the specific diversity functions they aim to optimize.

5.3 Experimental setting

To evaluate the performance of the examined recommendation strategies, we divide user interactions into a training and a test set, following an 80-20% split ratio. When evaluating the accuracy, we only focus on the recommendation list generated in the initial exploration step. Regarding diversity, we consider the complete set of recommendation lists produced across all exploration steps. Our approach also assumes that the entire item catalog is accessible to every user during the simulation.

To calculate the relevance score $\mathcal{R}(u, i)$, we employ a black-box model in the form of a neural network based on matrix factorization [27]. We fine-tune the latent factors of this model for each dataset. For EXPLORE, we use a value of $\alpha = 0.5$ in the Clayton copula. Additionally, we conduct hyperparameter tuning for this parameter, and it appears that it has no significant impact on the results (further details can be found in the Appendix).

We keep the length of the recommendation list, $\mathbf{L}$, fixed at 10, and vary the expected number of steps, $\mathbb{E}[\text{steps}]$, in the range of [5, 10, 20]. This allows us to devise a suitable value for the Weibull parameter λ to be used in the simulation experiments, according to Equation (7). To assess recommendation quality, we use standard metrics: *Hit-Ratio* (*HR*), *Precision*, and *Recall*. Our experimental results are the average of 20 independent trials, and we use the ANOVA test [17] to evaluate statistical significance. The code used in these experiments is made publicly accessible.[11]

Table 2: Diversity scores for $\mathbb{E}[\text{steps}] = 5$. Any best scores with a statistical significance $p < 0.05$ are highlighted in bold.

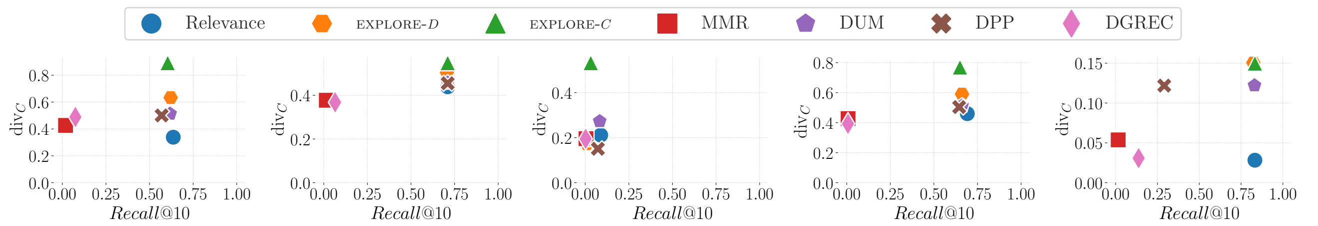
Dataset	Strategy	div_D	div_C	κ	$\Delta_{\bar{D}}$	$\Delta_{\bar{C}}$	$\Delta_{\mathbb{E}[\text{steps}]}$
Movielens-1M	Relevance	3.67	0.22	5.0	0.27	0.73	0.0
	EXPLORE-*D*	**4.91**	0.36	4.98	**0.03**	0.55	0.0
	EXPLORE-*C*	4.43	**0.71**	4.71	0.12	**0.11**	0.06
	MMR	3.96	0.29	4.57	0.22	0.64	0.09
	DUM	4.4	0.33	4.98	0.13	0.59	0.0
	DPP	4.59	0.31	4.99	0.09	0.61	0.0
	DGREC	3.36	0.37	4.49	0.33	0.54	0.1
Coat	Relevance	3.15	0.3	4.36	0.38	0.3	0.13
	EXPLORE-*D*	**3.48**	0.34	4.16	**0.31**	0.2	0.17
	EXPLORE-*C*	3.36	**0.35**	4.13	0.33	**0.18**	0.17
	MMR	2.43	0.26	3.54	0.52	0.39	0.29
	DUM	3.11	0.3	4.31	0.38	0.3	0.14
	DPP	3.28	0.31	4.33	0.35	0.27	0.13
	DGREC	2.2	0.24	3.2	0.56	0.44	0.36
KuaiRec-2.0	Relevance	0.76	0.13	4.81	0.81	0.74	0.04
	EXPLORE-*D*	**1.56**	0.11	3.54	**0.61**	0.78	0.29
	EXPLORE-*C*	1.08	**0.34**	4.09	0.73	**0.32**	0.18
	MMR	1.25	0.12	3.89	0.68	0.76	0.22
	DUM	0.83	0.17	4.8	0.79	0.66	0.04
	DPP	1.38	0.09	4.75	0.65	0.82	0.05
	DGREC	0.77	0.11	2.64	0.81	0.78	0.47
Netflix	Relevance	4.04	0.32	4.86	0.2	0.59	0.03
	EXPLORE-*D*	**4.62**	0.38	4.75	**0.09**	0.51	0.05
	EXPLORE-*C*	3.97	**0.6**	4.43	0.21	**0.22**	0.11
	MMR	3.56	0.3	4.19	0.3	0.61	0.16
	DUM	4.16	0.36	4.89	0.18	0.53	0.02
	DPP	4.38	0.34	4.88	0.13	0.56	0.02
	DGREC	3.0	0.26	3.74	0.41	0.66	0.25
Yahoo-R2	Relevance	0.66	0.02	**4.77**	0.87	0.77	**0.05**
	EXPLORE-*D*	**4.4**	**0.08**	4.49	**0.13**	**0.1**	0.1
	EXPLORE-*C*	4.38	**0.08**	4.47	**0.13**	**0.1**	0.11
	MMR	2.45	0.04	3.96	0.52	0.55	0.21
	DUM	4.38	0.07	4.72	**0.13**	0.21	0.06
	DPP	4.38	0.07	4.72	**0.13**	0.21	0.06
	DGREC	1.02	0.02	3.68	0.8	0.77	0.26

5.4 Results

Quality-diversity trade-off. We initiate our evaluation by assessing the performance of all our strategies in terms of recommendation quality and diversity. Figure 3 displays the scores for *Recall*@10 (on the x-axis) and coverage-based diversity (on the y-axis) across all five datasets. The figure shows that on all datasets, MMR and DGREC exhibit notably poor performance with respect to *Recall*@10. In contrast, the other strategies achieve significantly higher scores, with the Relevance baseline performing the best, which aligns with our expectations.

In terms of diversity, our method, EXPLORE-*C*, clearly outperforms the other strategies. It achieves a substantially higher diversity score while still delivering relevant recommendations. In fact, it strikes the best trade-off between diversity and relevance. Similar results, both in terms of diversity measures and other evaluation metrics, can be found in the Appendix for reference.

[11]https://anonymous.4open.science/r/EXPLORE-AA84/



(a) Movielens-1M (b) Coat (c) KuaiRec-2.0 (d) Netflix-Prize (e) Yahoo-R2

Figure 3: Trade-off between div_C and *Recall*@10 across all the datasets considered. The X-axis represents recommendation quality, while the Y-axis indicates the diversity score.

Best performing diversity strategy. Table 2 presents a comprehensive analysis of div_D, div_C and κ when $\mathbb{E}[\text{steps}] = 5$. Additionally, we report the deviations from the maximum diversity scores in terms of distance and coverage (reported in the Appendix), denoted as $\Delta_{\bar{D}}$ and $\Delta_{\bar{C}}$, along with $\Delta_{\mathbb{E}[\text{steps}]}$.

We observe that our strategy, either EXPLORE-*D* or EXPLORE-*C*, consistently outperforms the competitors in terms of both div_D and div_C across all datasets. We also show how these values deviate from the expected maximum values. Notably, on the Movielens-1M dataset, their scores are very close to their maxima. Our strategy achieves significantly higher scores than the competitors on all datasets, especially in terms of coverage.

Regarding the number of steps, as mentioned in Section 3, the relevance plays a fundamental role in our exploration process. Therefore, it is expected that our strategy performs slightly worse than other competitors, particularly the Relevance baseline. Nevertheless, our primary objective is to maximize recommendation diversity while maintaining relevance as high as possible. Additional details on the experiments are in the Appendix.

Ablation study. In our final investigation, we explore the advantages of combining both relevance and diversity through the copula function in Equation (10), in contrast to a simpler strategy that neglects relevance and relies on Equation (9).

Table 3 presents a summary of the results obtained for $\mathbb{E}[\text{steps}] = 10$. For each strategy, we provide the values for div_D, div_C, and actual steps κ. The scores are computed for two variants: one where relevance is included through the copula function (w) and another where it is ignored (w/o). The table also reports the differences in scores (Δ_w). We can observe that the combination has a positive effect both in terms of diversity and number of steps.

Table 3: Results obtained in terms of div_D, div_C and steps by including (w) and excluding (w/o) relevance from our recommendation strategies. Positive relative changes (Δ_w) are reported in bold.

	Strategy	Relevance	div_D	div_C	κ
Movielens-1M	EXPLORE-*D*	w	9.77	0.63	9.85
		w/o	6.66	0.53	6.73
		Δ_w	**+0.32**	**+0.16**	**+0.32**
	EXPLORE-*C*	w	8.84	0.89	9.7
		w/o	6.56	0.86	7.0
		Δ_w	**+0.26**	**+0.03**	**+0.28**
Coat	EXPLORE-*D*	w	6.73	0.51	8.02
		w/o	5.5	0.47	6.41
		Δ_w	**+0.18**	**+0.08**	**+0.2**
	EXPLORE-*C*	w	6.31	0.55	7.81
		w/o	5.18	0.49	6.34
		Δ_w	**+0.18**	**+0.11**	**+0.19**
KuaiRec-2.0	EXPLORE-*D*	w	3.06	0.17	6.86
		w/o	2.38	0.13	4.47
		Δ_w	**+0.22**	**+0.24**	**+0.35**
	EXPLORE-*C*	w	2.22	0.53	7.95
		w/o	2.01	0.49	6.04
		Δ_w	**+0.09**	**+0.08**	**+0.24**
Netflix	EXPLORE-*D*	w	9.03	0.59	9.24
		w/o	7.42	0.54	7.52
		Δ_w	**+0.18**	**+0.08**	**+0.19**
	EXPLORE-*C*	w	7.92	0.77	8.69
		w/o	6.71	0.75	7.38
		Δ_w	**+0.15**	**+0.03**	**+0.15**
Yahoo-R2	EXPLORE-*D*	w	8.71	0.15	8.73
		w/o	6.23	0.11	6.3
		Δ_w	**+0.28**	**+0.27**	**+0.28**
	EXPLORE-*C*	w	8.67	0.15	8.7
		w/o	6.25	0.11	6.31
		Δ_w	**+0.28**	**+0.27**	**+0.27**

6 CONCLUSION AND FUTURE WORK

In this study, we addressed recommendation diversity by introducing a user-behavior model where relevance drives engagement. We devised a recommendation strategy that optimizes the delivery of diverse knowledge to users in accordance with the underlying user behavior. The experimental analysis confirms that our approach is effective, yet it remains open to further enhancements. First, the behavioral model can be refined to encompass more sophisticated scenarios. These include alternative actions such as refreshing the list or guiding its composition, and incorporating dynamic adjustments to the weariness probability beyond temporal decay. Also, our model assumes that the underlying relevance score captures a user's actual interest in an item. However, the relevance score is computed through an algorithm that is not necessarily totally accurate. We can still adapt the user behavior model to account for such inaccuracy by incorporating a random discount factor associated with the relevance of each item in the list. Finally, the proposed strategy can be improved in several ways: for example, by integrating different distance measures or tailoring it to encompass additional metrics beyond diversity, such as serendipity or fairness.

REFERENCES

[1] Panagiotis Adamopoulos and Alexander Tuzhilin. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Trans. Intell. Syst. Technol.*, 2014. doi: 10.1145/2559952.

[2] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5), 2012. doi: 10.1109/TKDE.2011.15.

[3] Haifa Alharthi, Diana Inkpen, and Stan Szpakowicz. A survey of book recommender systems. *Journal of Intelligent Information Systems*, 51:139–160, 2018.

[4] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Optimal greedy diversity for recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2015. ISBN 9781577357384.

[5] James Bennett and Stan Lanning. The netflix prize. 2007. URL https://api.semanticscholar.org/CorpusID:9528522.

[6] Paul Black. Dads: The on-line dictionary of algorithms and data structures, 2020-09-17 2020.

[7] Dimitrios Bountouridis, Jaron Harambam, Mykola Makhortykh, Mónica Marrero, Nava Tintarev, and Claudia Hauff. Siren: A simulation framework for understanding the effects of recommender systems in online news environments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19. Association for Computing Machinery, 2019. doi: 10.1145/3287560.3287583.

[8] Robin Burke. *Hybrid Web Recommender Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-72079-9_12. URL https://doi.org/10.1007/978-3-540-72079-9_12.

[9] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1998. doi: 10.1145/290941.291025.

[10] Pablo Castells, Neil Hurley, and Saúl Vargas. *Novelty and Diversity in Recommender Systems*. Springer US, 2022. doi: 10.1007/978-1-0716-2197-4_16.

[11] Òscar Celma and Perfecto Herrera. A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08. Association for Computing Machinery, 2008. doi: 10.1145/1454008.1454038.

[12] Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18. Association for Computing Machinery, 2018. doi: 10.1145/3240323.3240370.

[13] Laming Chen, Guoxin Zhang, and Hanning Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18. Curran Associates Inc., 2018.

[14] D. G. CLAYTON. A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence. *Biometrika*, 1978. doi: 10.1093/biomet/65.1.141.

[15] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 540–550, 2022. doi: 10.1145/3511808.3557220. URL https://doi.org/10.1145/3511808.3557220.

[16] Zhaolin Gao, Tianshu Shen, Zheda Mai, Mohamed Reda Bouadjenek, Isaac Waller, Ashton Anderson, Ron Bodkin, and Scott Sanner. Mitigating the filter bubble while maintaining relevance: Targeted diversification with vae-based recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22. Association for Computing Machinery, 2022. doi: 10.1145/3477495.3531890.

[17] Ellen R Girden. *ANOVA: Repeated measures*. Number 84. Sage, 1992.

[18] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 53–62, 2020.

[19] Christian Hansen, Rishabh Mehrotra, Casper Hansen, Brian Brost, Lucas Maystre, and Mounia Lalmas. Shifting consumption towards diverse content on music streaming platforms. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21. Association for Computing Machinery, 2021. doi: 10.1145/3437963.3441775.

[20] Muhammad Haroon, Anshuman Chhabra, Xin Liu, Prasanta Mohapatra, Zubair Shafiq, and Magdalena E. Wojcieszak. Youtube, the great radicalizer? auditing and mitigating ideological biases in youtube recommendations. *ArXiv*, abs/2203.10666, 2022. URL https://api.semanticscholar.org/CorpusID:247594584.

[21] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 2015. doi: 10.1145/2827872.

[22] Naieme Hazrati and Francesco Ricci. Recommender systems effect on the evolution of users' choices distribution. *Information Processing & Management*, 59(1):102766, 2022. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2021.102766.

[23] Naieme Hazrati, Mehdi Elahi, and Francesco Ricci. Simulating the impact of recommender systems on the evolution of collective users' choices. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, HT '20. Association for Computing Machinery, 2020. ISBN 9781450370981. doi: 10.1145/3372923.3404812.

[24] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. 2004. doi: 10.1145/963770.963772.

[25] Neil Hurley and Mi Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM Trans. Internet Technol.*, 2011. doi: 10.1145/1944339.1944341.

[26] M.S.A. Khan, Abdul Khalique, and A. M. Abouammoh. On estimating parameters in a discrete weibull distribution. *IEEE Transactions on Reliability*, 38:348–350, 1989. URL https://api.semanticscholar.org/CorpusID:122132642.

[27] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009. doi: 10.1109/MC.2009.263.

[28] Dokyun Lee and Kartik Hosanagar. Impact of recommender systems on sales volume and diversity. In *International Conference on Interaction Sciences*, 2014. URL https://api.semanticscholar.org/CorpusID:2888230.

[29] Dokyun Lee and Kartik Hosanagar. How do recommender systems affect sales diversity? a cross-category investigation via randomized field experiment. *SSRN Electronic Journal*, 06 2017. doi: 10.2139/ssrn.2603361.

[30] Yile Liang, Tieyun Qian, Qing Li, and Hongzhi Yin. Enhancing domain-level and user-level adaptivity in diversified recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21. Association for Computing Machinery, 2021. doi: 10.1145/3404835.3462957.

[31] Chao Liu, Ryen W White, and Susan Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 379–386, 2010.

[32] Jian-Guo Liu, Kerui Shi, and Qiang Guo. Solving the accuracy-diversity dilemma via directed random walks. *Phys. Rev. E*, 2012. doi: 10.1103/PhysRevE.85.016118.

[33] Christian Matt, Thomas Hess, and Christian Weiß. The differences between recommender technologies in their impact on sales diversity. volume 4, 12 2013.

[34] P Novianti, S H Kartiko, and D Rosadi. Application of clayton copula to identify dependency structure of covid-19 outbreak and average temperature in jakarta indonesia. *Journal of Physics: Conference Series*, 1943(1):012154, jul 2021. doi: 10.1088/1742-6596/1943/1/012154.

[35] Derek O'Callaghan, Derek Greene, Maura Conway, Joe Carthy, and Pádraig Cunningham. Down the (white) rabbit hole: The extreme right and online recommender systems. *Social Science Computer Review*, 33, 2015.

[36] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables and stochastic processes*. 2002.

[37] Javier Parapar and Filip Radlinski. Diverse user preference elicitation with multi-armed bandits. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21. Association for Computing Machinery, 2021. doi: 10.1145/3437963.3441786.

[38] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72079-9. doi: 10.1007/978-3-540-72079-9_10.

[39] Hanyu Peng, Guanhua Fang, and Ping Li. Copula for instance-wise feature selection and rank. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 1651–1661. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/peng23a.html.

[40] Shruti Phadke, Mattia Samory, and Tanushree Mitra. Pathways through conspiracy: The evolution of conspiracy radicalization through engagement in online conspiracy discussions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 770–781, 2022.

[41] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09. AUAI Press, 2009. ISBN 9780974903958.

[42] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 131–141, 2020.

[43] Horst Rinne. *The Weibull Distribution: A Handbook (1st ed.)*. Chapman and Hall/CRC, 2008. doi: https://doi.org/10.1201/9781420087444.

[44] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. *Collaborative Filtering Recommender Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-72079-9_9. URL https://doi.org/10.1007/978-3-540-72079-9_9.

[45] Markus Schedl, Peter Knees, Brian McFee, Dmitry Bogdanov, and Marius Kaminskas. Music recommender systems. *Recommender systems handbook*, pages 453–492, 2015.

[46] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16. JMLR.org, 2016.

[47] Chaofeng Sha, Xiaowei Wu, and Junyu Niu. A framework for recommending relevant and diverse items. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16. AAAI Press, 2016.

[48] Barry Smyth and Paul McClave. Similarity vs. diversity. In *International Conference on Case-Based Reasoning*, 2001. URL https://api.semanticscholar.org/CorpusID:32908534.

[49] Harald Steck. Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18. Association for Computing Machinery, 2018. ISBN 9781450359016. doi: 10.1145/3240323.3240372.

[50] Zoltán Szlávik, Wojtek Kowalczyk, and Martijn Schut. Diversity measurement of recommender systems under different user choice models. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):369–376, Aug. 2021. doi: 10.1609/icwsm.v5i1.14116.

[51] Zeynep Tufekci. Youtube, the great radicalizer. *The New York Times*, 10(3):2018, 2018.

[52] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11. Association for Computing Machinery, 2011. doi: 10.1145/2043932.2043955.

[53] Prince Zizhuang Wang and William Yang Wang. Neural gaussian copula for variational autoencoder. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:202538985.

[54] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. Personalized news recommendation: Methods and challenges. *ACM Transactions on Information Systems*, 41(1):1–50, 2023.

[55] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 15–24, 2016.

[56] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S. Yu, and Taiqing Wang. Dgrec: Graph neural network for recommendation with diversified embedding generation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2023. doi: 10.1145/3539597.3570472.

[57] Sirui Yao, Yoni Halpern, Nithum Thain, Xuezhi Wang, Kang Lee, Flavien Prost, Ed H. Chi, Jilin Chen, and Alex Beutel. Measuring recommender system effects with simulated users. *ArXiv*, abs/2101.04526, 2021. URL https://api.semanticscholar.org/CorpusID:231583156.

[58] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 221–229, 2013.

[59] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11. Curran Associates Inc., 2011.

[60] Zhi Zeng and Ting Wang. Neural copula: A unified framework for estimating generic high-dimensional copula functions, 2022.

[61] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. Dgcn: Diversified recommendation with graph convolutional networks. In *Proceedings of the Web Conference 2021*, WWW '21. Association for Computing Machinery, 2021. doi: 10.1145/3442381.3449835.

[62] Dong Hong Zhu, Yawei Wang, and Ya Ping Chang. The influence of online cross-recommendation on consumers' instant cross-buying intention: The moderating role of decision-making difficulty. *Internet Res.*, 28:604–622, 2018. URL https://api.semanticscholar.org/CorpusID:49385888.

[63] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, 2005. doi: 10.1145/1060745.1060754.

[64] Morteza Zihayat, Anteneh Ayanso, Xing Zhao, Heidar Davoudi, and Aijun An. A utility-based news recommendation system. *Decision support systems*, 117: 14–27, 2019.

A APPENDIX

Complexity. In the following, we analyze the computational complexity of the proposed algorithm EXPLORE. Besides the (black-box) recommender system, the critical point is the generation of list $\mathbf{L}_t$ to be presented to users, where t is an exploration step in a session. Within an online setting, this list has to be dynamically generated and hence can represent a bottleneck. For the generation of $\mathbf{L}_t$, we need to consider Equations 9 and 10. By considering that $\mathcal{X}_t$ is computed incrementally, the cost of computing $\mathcal{T}_i$ in 9 is $O(td)$ when adopting div_D, and $O(|C|)$ when adopting div_C. Here, d is the computational cost associated with the Jaccard distance. In total, the worst-case cost for generating a list of k elements by considering n items is either $O(ndk^2)$ or $O(nk|C|)$.
We can observe the following. (1) The number n of items to consider could be large (in principle, the entire item catalog). However, since $\mathcal{T}_i$ is combined with $\mathcal{R}_i$ in Equation 10, we can filter out low-relevance items, as they will severely affect the value of $\mathcal{Z}_i$ due to the properties of the copula function. Notice also that, in a practical implementation, sampling strategies on portions of the catalog can also be devised to narrow the focus only on a subset of items. (2) The cost d for computing $d(i, j)$, for two generic items, can be $O(m)$, where m is the total number of users. To relieve this cost, the scores for popular items can be precomputed. Notice that in typical settings the distribution of items is heavy-tailed, thus we can expect that the number of distance scores to precompute is not intractably large.
Figure 6 shows that indeed our strategy exhibits comparable running times with those of other strategies, while significantly overcoming competitors such as MMR and DPP (that struggle especially with large datasets).

Experiments. We present additional experiments regarding the recommendation quality of the strategies, the diversity scores, and the run time required to provide the recommendation list.

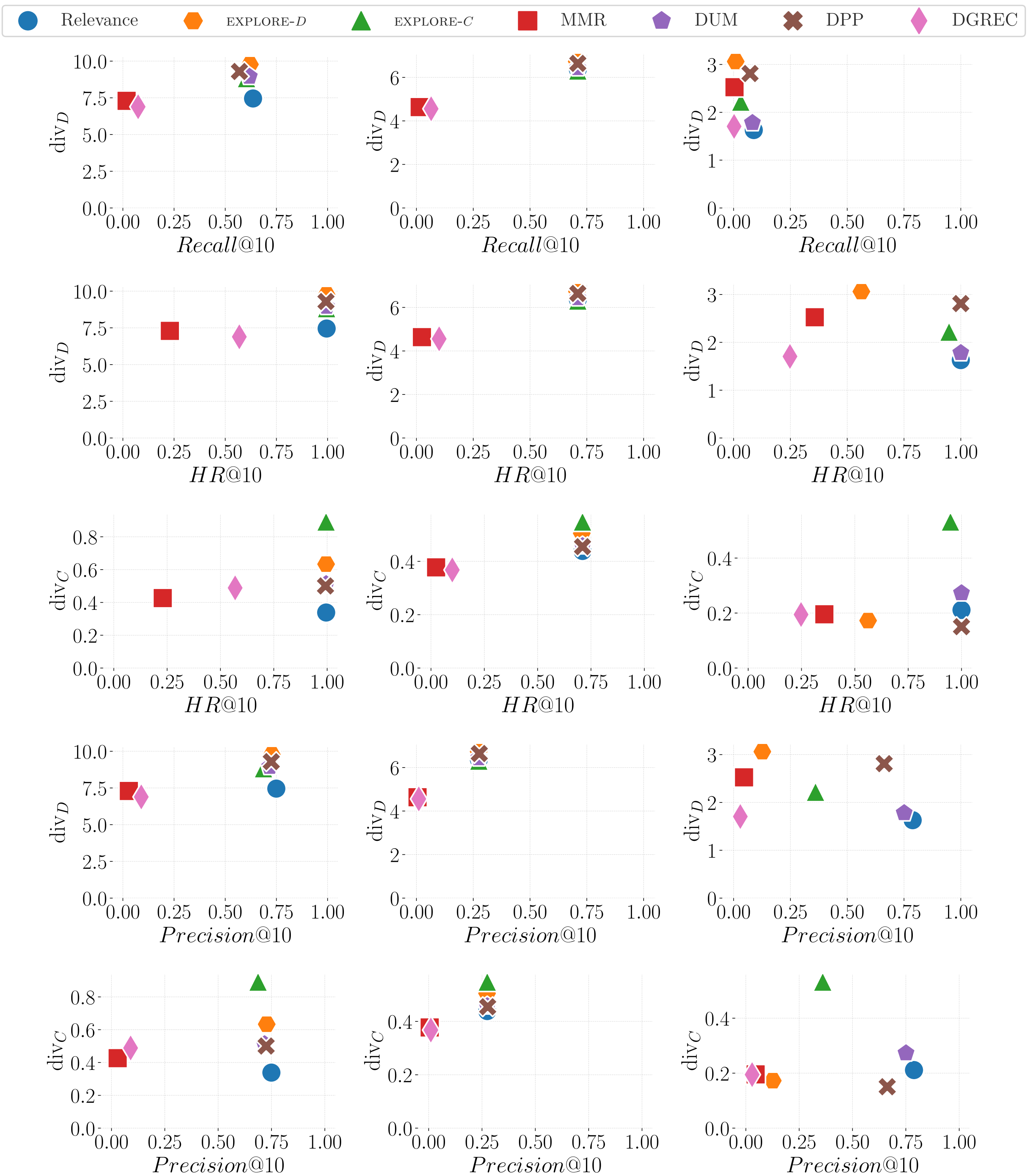
Figures 4 and 5 show the trade-off between recommendation quality and diversity, either in terms of div_D or div_C, and either $\{Recall, HR, Precision\}$@10. Further, Tables 4 and 5 report the scores in terms of div_D, div_C, κ, $\Delta_{\bar{D}}$, $\Delta_{\bar{C}}$, and $\Delta_{\mathbb{E}[\text{steps}]}$, obtained by varying the expected number of steps in the range [5, 10, 20].

As we can see, our algorithm, for both variants EXPLORE-D and EXPLORE-C, outperforms the competitors for most of the datasets, offering a significant improvement especially in terms of coverage.

Notably, despite the fact that in some settings EXPLORE-D obtains a slightly lower div_D score than DPP, it is crucial to consider the timing needed to provide the recommendation list, reported in Figure 6. As we can see, competitors such as MMR and DPP require a considerable amount of time to compute their recommended lists, in particular for the largest datasets. Our algorithm, instead, proves to be much more efficient, and its running time is basically constant over all the benchmarks.

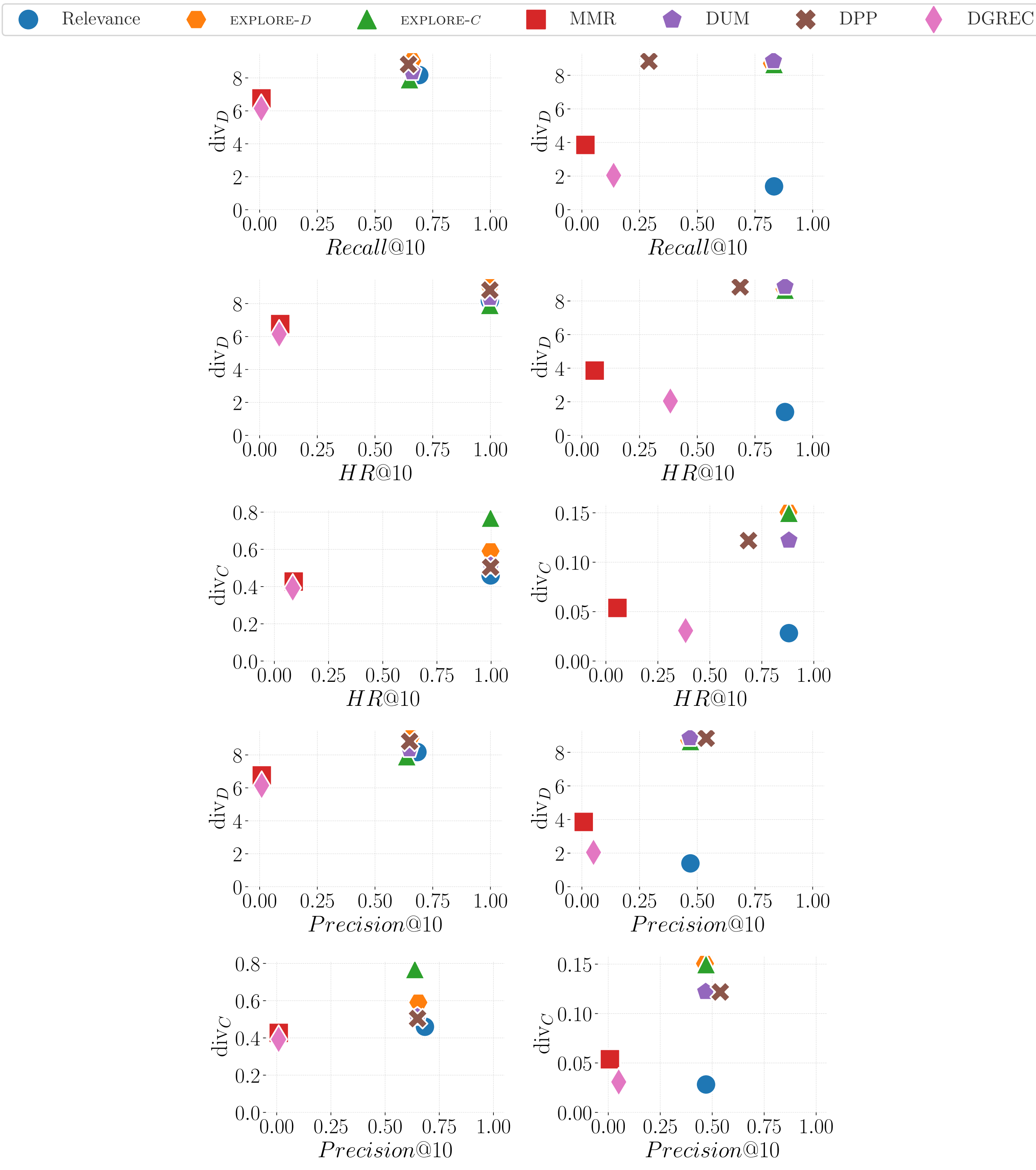
In Figure 7 we show the effect of tuning the α parameter of the Clayton copula function. As we can see, the parameter value does not affect the performance of the algorithm, thus proving EXPLORE to be free of hyper-parameter tuning.

Finally, Table 6 reports the maximum scores computed either in terms of distance ($\bar{D}$) or coverage ($\bar{C}$), while Table 7 summarizes the notation adopted throughout the paper.



(a) Movielens-1M **(b) Coat** **(c) KuaiRec-2.0**

Figure 4: Trade-off between either div_D or div_C and either *Recall*@10, *HR*@10, or *Precision*@10 over Movielens-1M, Coat and KuaiRec-2.0. The X-axis shows the recommendation quality while the Y-axis represents the diversity score.



(a) Netflix-Prize **(b) Yahoo-R2**

Figure 5: Trade-off between either div_D or div_C and either *Recall*@10, *HR*@10, or *Precision*@10 over Netflix-Prize and Yahoo-R2. The X-axis shows the recommendation quality while the Y-axis represents the diversity score.

Table 4: Results with $\mathbb{E}[\text{steps}] \in [5, 10, 20]$ over Movielens-1M, Coat and KuaiRec-2.0. Best scores with statistical significance $p < 0.05$ are in bold.

$\mathbb{E}[\text{steps}]$	Strategy	div_D	div_C	κ	$\Delta_{\bar{D}}$	$\Delta_{\bar{C}}$	$\Delta_{\mathbb{E}[\text{steps}]}$
5	Relevance	3.67	0.22	5.0	0.27	0.73	0.0
	EXPLORE-D	**4.91**	0.36	4.98	**0.03**	0.55	0.0
	EXPLORE-C	4.43	**0.71**	4.71	0.12	**0.11**	0.06
	MMR	3.96	0.29	4.57	0.22	0.64	0.09
	DUM	4.4	0.33	4.98	0.13	0.59	0.0
	DPP	4.59	0.31	4.99	0.09	0.61	0.0
	DGREC	3.36	0.37	4.49	0.33	0.54	0.1
10	Relevance	7.45	0.34	10.02	0.26	0.64	-0.0
	EXPLORE-D	**9.77**	0.63	9.85	**0.03**	0.32	0.02
	EXPLORE-C	8.84	**0.89**	9.7	0.12	**0.05**	0.03
	MMR	7.29	0.43	8.62	0.27	0.54	0.14
	DUM	8.95	0.51	10.03	0.11	0.45	-0.0
	DPP	9.29	0.5	9.99	0.08	0.46	0.0
	DGREC	6.89	0.49	8.92	0.31	0.47	0.11
20	Relevance	14.96	0.48	20.04	0.25	0.51	-0.0
	EXPLORE-D	**18.96**	0.86	19.4	**0.05**	0.12	0.03
	EXPLORE-C	16.81	**0.97**	19.76	0.16	**0.01**	0.01
	MMR	13.45	0.57	16.49	0.33	0.42	0.18
	DUM	17.98	0.7	**20.16**	0.1	0.29	**-0.01**
	DPP	18.6	0.71	20.02	0.07	0.28	-0.0
	DGREC	13.91	0.63	17.64	0.31	0.36	0.12

(a) Movielens-1M

$\mathbb{E}[\text{steps}]$	Strategy	div_D	div_C	κ	$\Delta_{\bar{D}}$	$\Delta_{\bar{C}}$	$\Delta_{\mathbb{E}[\text{steps}]}$
5	Relevance	3.15	0.3	4.36	0.38	0.3	0.13
	EXPLORE-D	**3.48**	0.34	4.16	**0.31**	0.2	0.17
	EXPLORE-C	3.36	**0.35**	4.13	0.33	**0.18**	0.17
	MMR	2.43	0.26	3.54	0.52	0.39	0.29
	DUM	3.11	0.3	4.31	0.38	0.3	0.14
	DPP	3.28	0.31	4.33	0.35	0.27	0.13
	DGREC	2.2	0.24	3.2	0.56	0.44	0.36
10	Relevance	6.38	0.44	8.64	0.36	0.35	0.14
	EXPLORE-D	6.73	0.51	8.02	0.33	0.25	0.2
	EXPLORE-C	6.31	**0.55**	7.81	0.37	**0.19**	0.22
	MMR	4.63	0.38	6.75	0.54	0.44	0.32
	DUM	6.44	0.46	8.65	0.36	0.32	0.13
	DPP	6.64	0.45	8.61	0.34	0.34	0.14
	DGREC	4.56	0.37	6.33	0.55	0.46	0.37
20	Relevance	12.23	0.59	16.36	0.39	0.33	0.18
	EXPLORE-D	12.85	0.7	15.48	0.36	0.21	0.23
	EXPLORE-C	12.09	**0.77**	15.36	0.4	**0.13**	0.23
	MMR	8.79	0.53	13.12	0.56	0.4	0.34
	DUM	12.63	0.62	16.78	0.37	0.3	0.16
	DPP	13.06	0.62	16.84	0.35	0.3	0.16
	DGREC	9.31	0.53	12.84	0.54	0.4	0.36

(b) Coat

$\mathbb{E}[\text{steps}]$	Strategy	div_D	div_C	κ	$\Delta_{\bar{D}}$	$\Delta_{\bar{C}}$	$\Delta_{\mathbb{E}[\text{steps}]}$
5	Relevance	0.76	0.13	4.81	0.81	0.74	0.04
	EXPLORE-D	**1.56**	0.11	3.54	**0.61**	0.78	0.29
	EXPLORE-C	1.08	**0.34**	4.09	0.73	**0.32**	0.18
	MMR	1.25	0.12	3.89	0.68	0.76	0.22
	DUM	0.83	0.17	4.8	0.79	0.66	0.04
	DPP	1.38	0.09	4.75	0.65	0.82	0.05
	DGREC	0.77	0.11	2.64	0.81	0.78	0.47
10	Relevance	1.63	0.21	9.6	0.79	0.71	0.04
	EXPLORE-D	**3.06**	0.17	6.86	**0.61**	0.77	0.31
	EXPLORE-C	2.22	**0.53**	7.95	0.72	**0.28**	0.2
	MMR	2.52	0.2	7.34	0.68	0.73	0.27
	DUM	1.78	0.27	9.59	0.77	0.63	0.04
	DPP	2.81	0.15	9.54	0.64	0.8	0.05
	DGREC	1.71	0.19	5.45	0.78	0.74	0.45
20	Relevance	3.62	0.32	19.02	0.77	0.65	0.05
	EXPLORE-D	**6.16**	0.26	13.83	**0.61**	0.71	0.31
	EXPLORE-C	4.55	**0.76**	15.09	0.71	**0.16**	0.25
	MMR	4.79	0.3	13.91	0.69	0.67	0.3
	DUM	3.86	0.39	19.09	0.75	0.57	0.05
	DPP	5.56	0.24	18.99	0.64	0.73	0.05
	DGREC	3.55	0.3	11.33	0.77	0.67	0.43

(c) KuaiRec-2.0

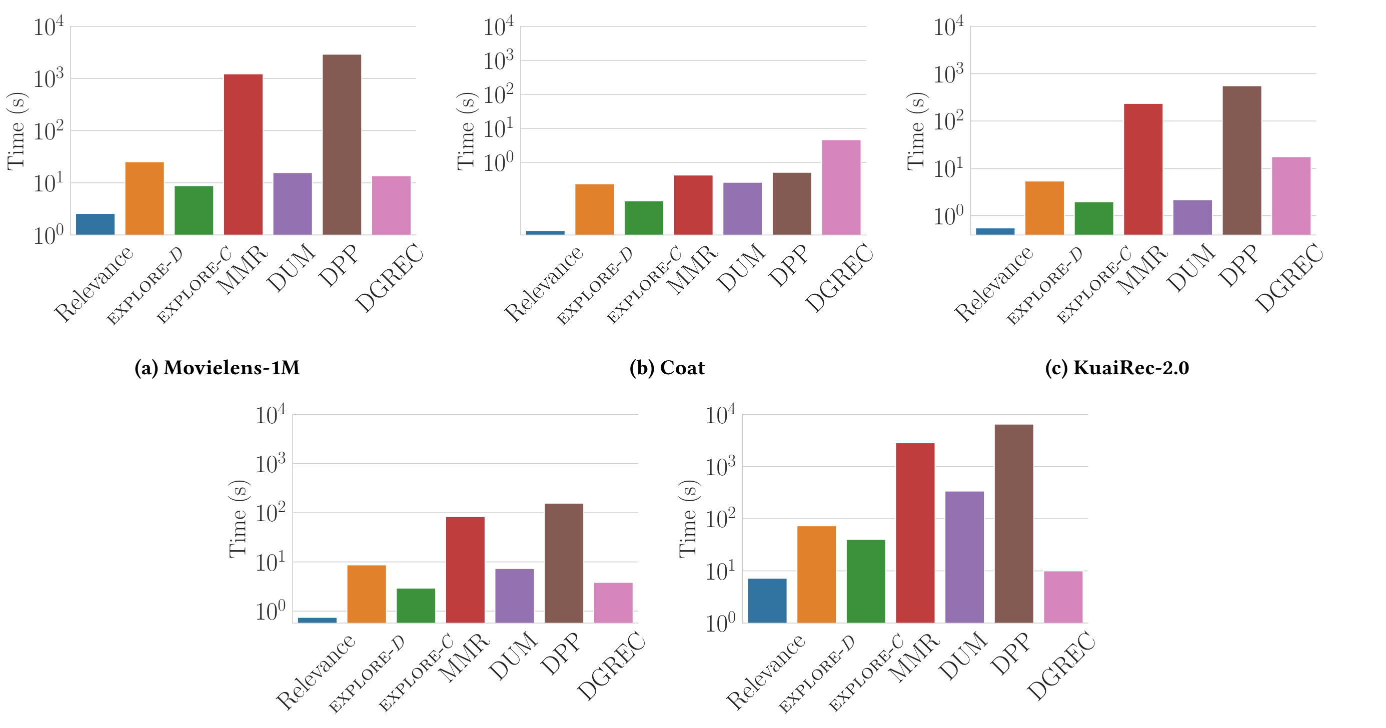
$\mathbb{E}[\text{steps}]$	Strategy	div_D	div_C	κ	$\Delta_{\bar{D}}$	$\Delta_{\bar{C}}$	$\Delta_{\mathbb{E}[\text{steps}]}$
5	Relevance	4.04	0.32	4.86	0.2	0.59	0.03
	EXPLORE-D	**4.62**	0.38	4.75	**0.09**	0.51	0.05
	EXPLORE-C	3.97	**0.6**	4.43	0.21	**0.22**	0.11
	MMR	3.56	0.3	4.19	0.3	0.61	0.16
	DUM	4.16	0.36	4.89	0.18	0.53	0.02
	DPP	4.38	0.34	4.88	0.13	0.56	0.02
	DGREC	3.0	0.26	3.74	0.41	0.66	0.25
10	Relevance	8.17	0.46	9.73	0.19	0.47	0.03
	EXPLORE-D	**9.03**	0.59	9.24	**0.1**	0.32	0.08
	EXPLORE-C	7.92	**0.77**	8.69	0.21	**0.12**	0.13
	MMR	6.76	0.43	7.94	0.33	0.51	0.21
	DUM	8.36	0.51	9.72	0.17	0.41	0.03
	DPP	8.82	0.5	9.73	0.12	0.43	0.03
	DGREC	6.15	0.39	7.44	0.39	0.55	0.26
20	Relevance	16.19	0.6	19.29	0.19	0.34	0.04
	EXPLORE-D	17.38	0.77	17.98	**0.13**	0.15	0.1
	EXPLORE-C	15.6	**0.87**	17.53	0.22	**0.04**	0.12
	MMR	12.79	0.56	15.34	0.36	0.38	0.23
	DUM	16.59	0.65	19.33	0.17	0.29	0.03
	DPP	**17.49**	0.66	19.36	**0.13**	0.27	0.03
	DGREC	12.11	0.53	14.59	0.4	0.42	0.27

(d) Netflix-Prize

Table 5: Results with $\mathbb{E}[\text{steps}] \in [5, 10, 20]$ over Yahoo-R2. Best scores with statistical significance $p < 0.05$ are in bold.

$\mathbb{E}[\text{steps}]$	Strategy	div_D	div_C	κ	$\Delta_{\bar{D}}$	$\Delta_{\bar{C}}$	$\Delta_{\mathbb{E}[\text{steps}]}$
5	Relevance	0.66	0.02	**4.77**	0.87	0.77	**0.05**
	EXPLORE-D	**4.4**	**0.08**	4.49	**0.13**	**0.1**	0.1
	EXPLORE-C	4.38	**0.08**	4.47	**0.13**	**0.1**	0.11
	MMR	2.45	0.04	3.96	0.52	0.55	0.21
	DUM	4.38	0.07	4.72	**0.13**	0.21	0.06
	DPP	4.38	0.07	4.72	**0.13**	0.21	0.06
	DGREC	1.02	0.02	3.68	0.8	0.77	0.26
10	Relevance	1.39	0.03	**9.49**	0.86	0.83	**0.05**
	EXPLORE-D	8.71	**0.15**	8.73	0.13	**0.14**	0.13
	EXPLORE-C	8.67	**0.15**	8.7	0.14	**0.14**	0.13
	MMR	3.86	0.05	7.36	0.62	0.71	0.26
	DUM	8.86	0.12	9.43	0.12	0.31	0.06
	DPP	8.84	0.12	9.41	0.12	0.31	0.06
	DGREC	2.04	0.03	7.35	0.8	0.83	0.26
20	Relevance	3.0	0.04	**18.81**	0.85	0.88	**0.06**
	EXPLORE-D	16.48	**0.28**	16.5	0.18	**0.17**	0.18
	EXPLORE-C	16.43	**0.28**	16.46	0.18	**0.17**	0.18
	MMR	5.9	0.07	13.89	0.71	0.79	0.31
	DUM	17.59	0.19	18.73	0.12	0.44	**0.06**
	DPP	17.6	0.19	18.74	0.12	0.44	**0.06**
	DGREC	3.92	0.04	14.56	0.8	0.88	0.27

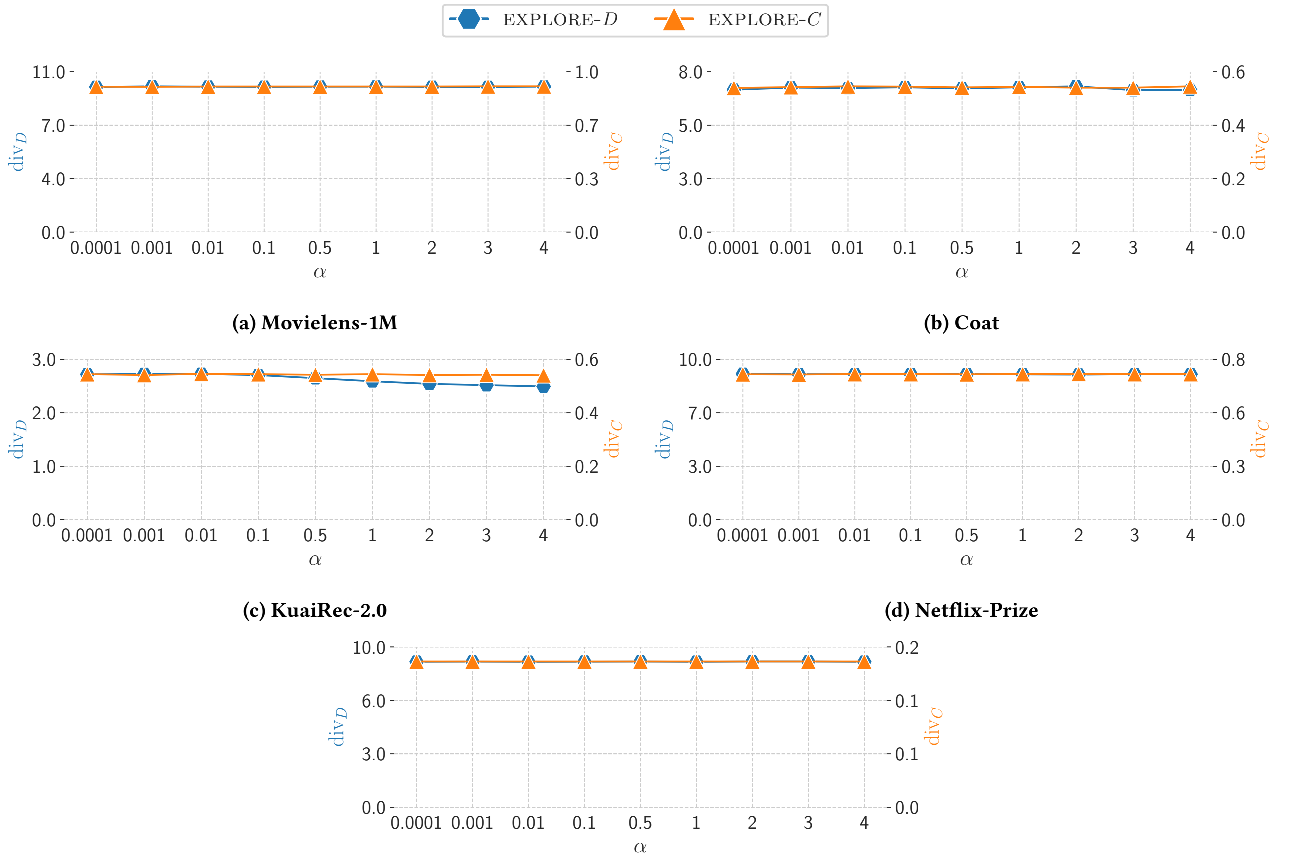
(a) Yahoo-R2



(a) Movielens-1M **(b) Coat** **(c) KuaiRec-2.0**

(d) Netflix-Prize **(e) Yahoo-R2**

Figure 6: Timing for producing a recommendation list. The X-axis reports the strategies, while the Y-axis shows the recommendation time (in seconds).



(a) Movielens-1M

(b) Coat

(c) KuaiRec-2.0

(d) Netflix-Prize

(e) Yahoo-R2

Figure 7: Effects of tuning the α parameter fixing $\mathbb{E}[\text{steps}] = 10$. The X-axis represents different values of α, while the Y-axes report values of div_D (left) and of div_C (right).

Table 6: Maximum scores in terms of diversity and coverage per dataset, by varying the expected number of steps.

Dataset	$\mathbb{E}[\text{steps}]$	$\bar{D}$	$\bar{C}$
Movielens-1M	5	5.05	0.80
	10	10.05	0.93
	20	20.03	0.98
Coat	5	5.05	0.43
	10	10.05	0.68
	20	20.03	0.89
KuaiRec-2.0	5	3.97	0.50
	10	7.87	0.73
	20	15.63	0.91
Netflix-Prize	5	5.05	0.77
	10	10.05	0.87
	20	20.03	0.91
Yahoo-r2	5	5.05	0.09
	10	10.05	0.17
	20	20.03	0.34

Table 7: Notation

$\mathcal{U}$	Set of users
$\mathcal{I}$	Set of items
C	Set of categories
$\mathcal{R}(u, i)$	Relevance score of the item i for the user u
$\mathcal{S}$	Recommendation strategy
$\mathbf{x}_i$	Users vector of item i
$\mathbf{y}_i$	Categories vector of item i
$d(i, j)$	Weighted Jaccard distance between two items i, j, either in terms of users or categories
div_D	Diversity score in terms of distance
div_C	Diversity score in terms of coverage
$\mathbf{L}_t$	Recommendation list produced at step t
k	Size of the recommendation lists
q_i	Probability for the user to be interested in the item i
p_i	Probability for the user to select the item i
η_t	Weariness probability, i.e., probability for the user to lose interest at step t
Q_t	Quitting probability, i.e., probability to quit the exploration process at step t
Q_T	Overall quitting probability, i.e., probability to quit the exploration process at any step
$\mathcal{X}$	Final user interactions set
κ	Actual number of steps performed by the user at the end of the exploration process
$\mathbb{E}[\text{steps}]$	Expected number of steps to be performed by the user at the end of the exploration process