Sequential Diversification with Provable Guarantees

Honglian Wang
KTH Royal Institute of Technology
Stockholm, Sweden
honglian@kth.se

Sijing Tu
KTH Royal Institute of Technology
Stockholm, Sweden
sijing@kth.se

Aristides Gionis
KTH Royal Institute of Technology
Stockholm, Sweden
argioni@kth.se

Abstract

Diversification is a useful tool for exploring large collections of information items. It has been used to reduce redundancy and cover multiple perspectives in information-search settings. Diversification finds applications in many different domains, including presenting search results of information-retrieval systems and selecting suggestions for recommender systems.

Interestingly, existing measures of diversity are defined over *sets* of items, rather than evaluating *sequences* of items. This design choice comes in contrast with commonly-used relevance measures, which are distinctly defined over sequences of items, taking into account the ranking of items. The importance of employing sequential measures is that information items are almost always presented in a sequential manner, and during their information-exploration activity users tend to prioritize items with higher ranking.

In this paper, we study the problem of *maximizing sequential diversity*. This is a new measure of *diversity*, which accounts for the *ranking* of the items, and incorporates *item relevance* and *user behavior*. The overarching framework can be instantiated with different diversity measures, and here we consider the measures of *sum diversity* and *coverage diversity*. The problem was recently proposed by Coppolillo et al. [11], where they introduce empirical methods that work well in practice. Our paper is a theoretical treatment of the problem: we establish the problem hardness and present algorithms with constant approximation guarantees for both diversity measures we consider. Experimentally, we demonstrate that our methods are competitive against strong baselines.

CCS Concepts

• Information systems → Information retrieval; **• Theory of computation → Approximation algorithms analysis**.

Keywords

Diversification, Ranking algorithms, Approximation algorithms

ACM Reference Format:
Honglian Wang, Sijing Tu, and Aristides Gionis. 2025. Sequential Diversification with Provable Guarantees. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM '25), March 10–14, 2025, Hannover, Germany*. ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3701551.3703531

WSDM '25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03
https://doi.org/10.1145/3701551.3703531

1 Introduction

Diversification has been widely adopted in information retrieval and recommendation systems to facilitate information exploration, that is, to present users with content that is not only relevant, but also novel and diverse. Diversification benefits both users and service providers [26]: it enhances user experience by mitigating the filter-bubble effect and promoting a fair representation of perspectives, while helping service providers avoid over-exposure of popular items and cater to broader information needs.

Numerous diversification approaches have been proposed [26], including methods that aim to balance the tradeoff between *diversity* and *relevance* [5, 16]. However, most existing approaches have two shortcomings. First, they assume a fixed number of items to be selected, overlooking the fact that users may lose interest and terminate their information-seeking task early. Second, many approaches formulate the diversification task as a *set-selection problem*, rather than a *ranking problem*, thereby ignoring the importance of the ordering of selected items.

These shortcomings arise from failing to incorporate the users' engagement behavior. Users may exit due to tiredness or disinterest, making it challenging for the system to determine beforehand how many content items the users will examine. Additionally, the ranking of selected items is crucial, as users are more likely to examine items that are higher in the ranking. Note that many diversification methods employ greedy strategies [2, 5, 6, 10, 24], which implicitly provide a ranking of the items. However, such a ranking is not an integral part of the problem definition. It is not designed to meet any desirable properties and is merely a by-product of the algorithmic solution.

Our contributions. In this paper, we build on the setting of *sequential diversity*, introduced recently by Coppolillo et al. [11], aiming to address the above-mentioned shortcomings. The sequential-diversity setting defines a new diversity measure, which accounts for the *ranking* of items, and it incorporates *item relevance* together with *user behavior*. While the work of Coppolillo et al. [11] presents practical methods for this problem, our paper offers a rigorous analysis and algorithms with provable guarantees.

The sequential-diversity setting is formalized as follows. We consider that items are presented to users in a specific order and users examine them sequentially. We define the *continuation probability* of an item for a user as the probability that the user will *accept* the item and continue *examining* the next item in the sequence. We assume that continuation probabilities are related to the *item relevance* for a given user, and thus can be learned from observed data.

It follows that the model is *stochastic*, with the number of items examined being a random variable depending on the ranking of items and their continuation probabilities. The *sequential diversity* of an ordered sequence is defined as the *expected diversity* of the items accepted by the user, with the expectation taken over the continuation probabilities. Our objective is to maximize sequential diversity by computing an optimal ranking of items, which we refer to as *sequential-diversity maximization*. This model effectively captures the interplay between relevance, diversity, and user engagement, as achieving high sequential diversity requires rankings that incorporate *both diverse and relevant items* to maintain user engagement with the platform.

The model needs to be instantiated with a diversity measure for item sets, and any standard measure can be used. In this paper, we consider two commonly-used diversity measures: *pair-wise sum diversity* [5] and *coverage diversity* [4, 24, 28]. For coverage diversity, we prove the problem is *ordered-submodular* [22], and thus, existing techniques can be applied.

For pair-wise sum diversity measure and its corresponding maximization problem (MaxSSD, Problem 1), we need to develop novel methods. We establish the computational hardness of the problem and present algorithms with constant-factor approximation guarantees. Our techniques leverage connections to an *ordered* variant of the *Hamiltonian path* problem (Section 5), which seeks to maximize the sum of edge weights in the Hamiltonian path, where the edge weights depend on their order in the path. Our algorithms are grounded on the observation that the sequential diversity score is largely determined by the top items in the ranking; thus, we prioritize optimizing the selection and positioning of these top items.

Depending on the values of the continuation probabilities, we introduce two distinct algorithms. When continuation probabilities are relatively small, e.g., constant values that are smaller than 1, we design an algorithm that focuses on selecting the top-τ items, where the value of τ controls the trade-off between runtime and approximation quality. Notably, when $\tau = 2$, our approach aligns with a simple greedy approach, and hence, we provide the approximation ratio for this greedy approach. In contrast, when continuation probabilities are arbitrarily close to 1, it's not just the first few items that affect the sequential diversity score; rather, more items ranked toward the front impact the score. To address this case, we propose a second algorithm using a greedy matching approach on the set of all items, and offers a constant approximation guarantee.

In our experiments, we introduce metrics for user satisfaction and engagement. We experimentally compare our methods against baselines and demonstrate their effectiveness in finding high-quality solutions. Due to space limitations, proofs of our theoretical results and a more extensive experimental evaluation are provided in the appendix.

2 Related work

Content diversification is commonly formulated as an optimization problem [2, 4–10, 16, 22, 24]. Two common diversity functions are coverage diversity and pairwise-sum diversity.

Agrawal et al. [2] introduce the concept of user coverage, aiming to recommend a relevant and diverse set while maximizing the number of users who encounter at least one relevant document. Ashkan et al. [4] and Puthiya et al. [24] explore taxonomy coverage, defining diversity as the number of topics or users covered by the selected set of items. These studies formulate their objective as monotone submodular functions and apply greedy strategies with provable guarantees.

For pairwise sum diversity, Carbonell and Goldstein [6] introduce the *marginal maximal relevance* (MMR), which is a function that combines the pairwise distance and the relevance of the selected items to generate a ranking via greedily selection. Borodin et al. [5] and Gollapudi and Sharma [16] investigate the notion of *max-sum diversity* (MSD) by combining sum of pairwise distance with a submodular relevance function. Subsequent works on the MSD problem apply convex-programming [7] and local-search [8, 9] techniques to achieve better approximation ratios. Chen et al. [10] and Gan et al. [13] use a *determinantal point process* (DPP) model by maintaining a kernel matrix whose off-diagonal entries measure the similarity between items. Chen et al. [10] use the maximum a posterior (MAP) inference for DPP and propose a greedy MAP inference algorithm for efficient computation.

The main limitation of the aforementioned studies is the assumption that users consider all results equally, without accounting for the order of items Technically, although a greedy heuristic can produce an ordered sequence [2, 5, 6, 10, 24], the objective value does not depend on item ranking.

Some studies incorporate the ranking of items into their models. Ashkan et al. [3, 4] propose the *diversity-weighted utility maximization* (DUM) approach which ranks items solely based on their utilities (relevance scores, in our setting). Unlike DUM, our method integrates relevance and diversity directly into the ranking. Ieong et al. [20] and Tang et al. [25] address advertisement allocation in a sequential setting similar to ours, where users sequentially view content and exit with fixed probabilities. However, they solve a distinct set of problems.

The most closely related works to our problem are by Coppolillo et al. [11] and Kleinberg et al. [22]. Coppolillo et al. [11] introduce a user-behavior model that simulates user interactions with recommendation systems and propose the EXPLORE algorithm to maximize their proposed diversity measures. Their framework allows users to interact with multiple items at each time step and accept one of them. Moreover, they do not provide any provable guarantees of their approaches. Kleinberg et al. [22] assumes that users' patience decays as they progress through a result list. They propose an *ordered-submodular* coverage function and prove that a greedy algorithm achieves a $\frac{1}{2}$-approximation; this result extends to our problem when the coverage function is selected as the diversity measure. When pairwise distance is used as the diversity measure, we develop novel methods to approach our problem.

Recent works have applied deep-learning techniques to content diversification [1, 19, 27, 29], typically incorporating diversification as part of a broader optimization task. In contrast, our focus is on defining sequential diversity as a standalone optimization problem and developing methods with provable guarantees. Notably, our framework can leverage deep-learning approaches to learn continuation probabilities. Consequently, we view deep-learning approaches as complementary to our framework.

3 Preliminaries

We consider a system where users interact with distinct content items, such as videos or articles, presented in an ordered sequence. Let $U = \{1, \ldots, n\}$ represent the set of items. For a given user u, each item $i \in U$ has an associated *continuation probability* p_i, representing the probability that, upon examining item i, the user u *accepts* this item and continues examining subsequent items. Conversely, $1 - p_i$ is the probability of terminating the session and quit the system after examining item i.

We assume that p_i is determined by the *relevance* of item i for user u and is provided as input to the problem. While estimating p_i is an interesting task, it lies outside the scope of this work. In our empirical evaluation, we explore simple methods that map relevance scores to continuation probabilities.

The user examines items in U sequentially, following an order determined by the permutation $\pi : U \to U$, where $\pi(i)$ denotes the i-th item in the sequence. Let $O_{(\pi)} = (\pi(i))_{i=1}^n$ represent the ordered sequence defined by π; for simplicity, we omit the subscript π and write $O = O_{(\pi)}$. We denote by $O_k = (\pi(i))_{i=1}^k$ the k-prefix of O, i.e., the subsequence of O consisting of the first k items. Equivalently, we write $O_k \sqsubseteq O$.

After examining item $\pi(i)$, the user either *accepts* it and proceeds to examine the next item with probability $p_{\pi(i)}$, or rejects it and quits the system immediately.

Given an ordered sequence of items O, let us denote by $A \sqsubseteq O$ the ordered sequence of items the user examines and *accepts* before quitting. Notice that A is a random variable. The probability of the user accepting exactly the items of O_k is $\Pr(A = O_k)$. Let $\mathcal{D}(\cdot)$ denote a *diversity function* measuring the diversity of the accepted items. As we will discuss shortly, our goal is to maximize the expected diversity score $\mathcal{D}(A)$.

In our paper, an algorithm ALG achieves an α-approximation ($\alpha \leq 1$) if, for any instance I, it guarantees $\text{ALG}(I) \geq \alpha\, \text{OPT}(I)$.

3.1 Problem definition

We first introduce the *sequential-diversity* objective and then formalize the *maximization* problem we consider.

Definition 1 (Sequential diversity ($\mathcal{S}$)). *Let $U = \{1, \ldots, n\}$ be a finite set of n distinct items, and $p_1, \ldots, p_n$ be continuation probabilities assigned to each item. Let $O = (\pi(i))_{i=1}^n$ be an ordered sequence of U according to a permutation π. Let $\mathcal{D}(A)$ be a diversity function of the items in A. The* sequential diversity *of O, denoted by $\mathcal{S}(O)$, is defined as the expectation of the diversity of the items that the user accepts before quitting, i.e., $\mathcal{S}(O) = \mathbb{E}_{A \sqsubseteq O}[\mathcal{D}(A)]$.*

The expectation in Definition 1 is taken over the probability that the user *accepts* exactly $O_k \sqsubseteq O$, where $k = 0, 1, \ldots, n$.

We will work with two commonly-used diversity functions. The first is the *pair-wise sum diversity*: for a distance function $d : U \times U \to \mathbb{R}_{\geq 0}$, the diversity function $\mathcal{D}_+(A) = \sum_{i,j \in A} d(i, j)$ sums all pair-wise distances of items in U.

Accordingly, we define the *sequential sum diversity* objective.

Definition 2 (Sequential sum diversity ($\mathcal{S}_+$)). *The* sequential sum diversity *of a sequence O, denoted by $\mathcal{S}_+(O)$, is defined as in Definition 1, with diversity function $\mathcal{D} = \mathcal{D}_+$, i.e.,*

$$\mathcal{S}_+(O) = \mathbb{E}_{A \sqsubseteq O}[\mathcal{D}_+(A)] = \mathbb{E}_{A \sqsubseteq O}\left[\sum\nolimits_{i,j \in A} d(i, j)\right]. \tag{3.1}$$

Next, we define the problem MaxSSD of finding an ordering of items in U that maximizes the sequential sum diversity objective.

Problem 1 (Maximizing sequential sum diversity (MaxSSD)). *Given a finite set $U = \{1, \ldots, n\}$ of n distinct items and associated continuation probabilities $p_1, \ldots, p_n$, find an ordering O^* of the items in U that maximizes the sequential sum diversity objective, i.e.,*

$$O^* = \arg\max\nolimits_{O=\pi(U)} \mathcal{S}_+(O). \tag{3.2}$$

In Definition 2, the items in U can be used to define a *complete weighted graph*, with edge weights being the distances between the items. We make heavy use of this observation as we employ *graph-theoretic ideas*, such as *Hamiltonian paths*, *graph matchings*, etc.

The second (alternative) notion of diversity is based on *coverage* [4, 24]. Here we assume that each item $i \in U$ is associated with a set of attributes $X(i) \subseteq \mathcal{X}$, where $\mathcal{X}$ is the set of all attributes. For example, in a move dataset, $\mathcal{X}$ is the set of all movie genres, and $X(i)$ the genres of movie i. The *coverage diversity* of a sequence A is then defined as the set of attributes of all items in A, i.e., $\mathcal{D}_c(A) = \bigcup_{i \in A} X(i)$.

Analogously to Problem 1, we define the problem of *maximizing the sequential coverage diversity* (MaxSCD, Problem 2).

Definition 3 (Sequential coverage diversity ($\mathcal{S}_c$)). *The* sequential coverage diversity *of a sequence O, denoted by $\mathcal{S}_c(O)$, is defined as in Definition 1, with diversity function $\mathcal{D} = \mathcal{D}_c$, i.e.,*

$$\mathcal{S}_c(O) = \mathbb{E}_{A \sqsubseteq O}[\mathcal{D}_c(A)] = \mathbb{E}_{A \sqsubseteq O}\left[\bigcup\nolimits_{i \in A} X(i)\right]. \tag{3.3}$$

Problem 2 (Maximizing sequential coverage diversity (MaxSCD)). *Given a finite set $U = \{1, \ldots, n\}$ of n distinct items and associated probabilities $p_1, \ldots, p_n$, find an ordering O^* of the items in U that maximizes the sequential coverage diversity objective, i.e.,*

$$O^* = \arg\max\nolimits_{O=\pi(U)} \mathcal{S}_c(O). \tag{3.4}$$

3.2 Reformulation and complexity

We start our analysis with an observation that simplifies our objective. Observation 1 stems from the fact that the probabilities assigned to all items are independent.

To simplify our calculations, we use the notation $p_{O_i} = \prod_{t=1}^i p_{\pi(t)}$ and $d(\pi(i+1), O_i) = \sum_{t=1}^i d(\pi(i+1), \pi(t))$.

Observation 1. *The probability that the user does not accept any item is $\Pr(A = O_0) = 1 - p_{\pi(1)}$; the probability that the user quits after accepting the first k items is $\Pr(A = O_k) = \prod_{i=1}^k p_{\pi(i)}(1 - p_{\pi(k+1)})$, if $k \leq n - 1$; and the probability that the user accepts all items is $\Pr(A = O_n) = \prod_{i=1}^n p_{\pi(i)}$.*

Based on Observation 1, we reformulate Equation (3.1).

Lemma 1. *The sequential sum diversity objective in Equation (3.1) can be reformulated as*

$$\mathcal{S}_+(O) = \sum\nolimits_{i=1}^{n-1} p_{O_{i+1}} d(\pi(i+1), O_i).$$

We give a concrete example of how the ordering of the items influences the sequential sum diversity $\mathcal{S}_+$.

Example 1. *Let $U = \{u_1, u_2, u_3\}$, where $d(u_1, u_2) = 0.3$, $d(u_1, u_3) = 1$ and $d(u_2, u_3) = 1$. Let $p_{u_1} = p_{u_2} = 1$ and $p_{u_3} = 0$. The sequential sum diversity scores differ according to the order of the items, specifically:*

- $\mathcal{S}_+((u_1, u_2, u_3)) = \mathcal{S}_+((u_2, u_1, u_3)) = 0.3$, *as the user accepts both u_1 and u_2.*
- $\mathcal{S}_+((u_1, u_3, u_2)) = \mathcal{S}_+((u_2, u_3, u_1)) = 0$, *as the user only accepts u_1 or u_2.*
- $\mathcal{S}_+((u_3, u_1, u_2)) = \mathcal{S}_+((u_3, u_2, u_1)) = 0$, *as the user directly quits the system after examining u_3.*

Complexity. We can show that the problem of maximizing sequential sum diversity is **NP**-hard.

Theorem 1. *MaxSSD is* **NP***-hard, even when all p_i are equal.*

This result indicates our problem is non-trivial, and approximation algorithms are required to guarantee high-quality orderings.

4 Sequential coverage diversity

In this section, we show that the problem of maximizing sequential coverage diversity, MaxSCD, is ordered submodular [22].

Definition 4 (Ordered submodularity [22])**.** *A sequence function f is ordered-submodular if for all sequence X and Y, the following property holds for all elements s and $\bar{s}$:*

$$f(X||s) - f(X) \geq f(X||s||Y) - f(X||\bar{s}||Y) \tag{4.1}$$

where $||$ denotes the concatenation of sequences and elements.

Ordered submodularity is introduced by Kleinberg et al. [22], who extend the concepts of monotonicity and submodularity from set functions to sequence functions.

Observation 2. *The sequential coverage diversity function $\mathcal{S}_c(\cdot)$ is ordered-submodular.*

As proved by Kleinberg et al. [22], a simple greedy algorithm provides a $1/2$ approximation for an ordered submodular function, and thus, for MaxSCD. On the other hand, sequential sum diversity is not ordered submodular.

Observation 3. *The sequential sum diversity function $\mathcal{S}_+(\cdot)$ is not ordered-submodular.*

In the remainder of our paper, we focus on the technically more intriguing MaxSSD problem. Nonetheless, it is important to highlight that our definition of the MaxSCD problem, as well, provides a novel perspective to recommender systems and information-retrieval applications.

5 Ordered Hamiltonian path

In the previous section, we establish that MaxSSD is **NP**-hard through a reduction from the well-known *clique problem* [15], highlighting the inherent complexity of directly solving the problem. Furthermore, we observe that MaxSSD is not ordered submodular, precluding the use of standard *greedy* approaches for constant-factor approximations.

To address this, we introduce a new problem, *maximum ordered Hamiltonian path* (MaxOHP), and show in Theorem 2 that MaxSSD can be reduced to MaxOHP with only a constant-factor approximation loss. This reduction allows us to focus on MaxOHP, a more tractable problem, for which we propose approximation algorithms in subsequent sections.

Definition 5 (Ordered Hamiltonian path ($\mathcal{H}$))**.** *We are given a finite set $U = \{1, \ldots, n\}$ of n distinct items, a distance function $d(\cdot, \cdot)$, and probabilities $\{p_1, \ldots, p_n\}$ assigned to each item $i \in U$. Let $O = (\pi(i))_{i=1}^n$ be the ordered sequence of items of U according to an order π. The ordered Hamiltonian path, denoted by $\mathcal{H}$, is defined as*

$$\mathcal{H}(O) = \sum\nolimits_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1)), \tag{5.1}$$

where $W_{O_i} = \sum_{j=i+1}^n p_{O_j}$.

Note that the *ordered* Hamiltonian-path differs significantly from the classic Hamiltonian-path [17], where the edge weights do not depend on the *order* in which edges are placed. In contrast, in the ordered Hamiltonian-path, the edge weights $d(\pi(i), \pi(i+1))$ are multiplied by a coefficient W_{O_i}, which depends on the order π as well as on the probabilities of all the items in U.

Next, we define the problem of maximizing the ordered Hamiltonian path (MaxOHP).

Problem 3 (MaxOHP)**.** *We are given a finite set $U = \{1, \ldots, n\}$ of n distinct items, a distance function $d(\cdot, \cdot)$, and probabilities $p_1, \ldots, p_n$ assigned to each item $i \in U$. The goal is to find an order O^* of the items in U so as to maximize the $\mathcal{H}$ objective, that is,*

$$O^* = \arg\max_{O=\pi(U)} \mathcal{H}(O),$$

where $W_{O_i} = \sum_{j=i+1}^n p_{O_j}$.

The main result of this section is the following.

Theorem 2. *Let $p_i \in [a, b]$, with $0 < a < b < 1$. An α-approximation solution for MaxOHP is a $\frac{a(1-b)}{2b(1-a)}\alpha$-approximation for MaxSSD.*

Corollary 1. *Consider the special case where $p_i = p$ for all $i \in U$. An α-approximation solution for MaxOHP is a $\frac{\alpha}{2}$-approximation for MaxSSD.*

6 Uniform continuation probabilities

In this section, we address the case where all continuation probabilities p_i are equal, and we devise approximation algorithms for MaxSSD under this special case. Specifically, we assume p may vary as a function of n and can asymptotically approach 0 or 1, excluding the trivial cases of $p = 0$ or $p = 1$.

We analyze three regimes for p, and design algorithms that achieve constant-factor approximations for MaxSSD in each regime. Due to space constraints, two regimes are presented in the main content, with the third deferred to the appendix.

As outlined earlier, we reformulate the problem using the intemediate problem MaxOHP, which then leads to a solution for MaxSSD. Recall that for an order O, the ordered Hamiltonian-path objective is defined as $\mathcal{H}(O) = \sum_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1))$, where $W_{O_i} = \sum_{j=i+1}^n p_{O_j}$. In the uniform probability setting, since $p_i = p$ for all i, the coefficient W_{O_i} is entirely determined by its position in the order, rather than the actual items being ranked. In this case,

$$W_{O_i} = \sum\nolimits_{j=i+1}^{n} p^j = \frac{p^{i+1} - p^{n+1}}{1-p}. \tag{6.1}$$

Algorithm 1: Best-τ items (BτI) algorithm

Input: Finite set U, integer τ, probabilities p_i, diversity function d
Output: An ordered sequence of U according to π, $O_n \leftarrow \pi(U)$

if p_i *are uniform* **then**
  $O_\tau \leftarrow$ Solution of Equation (6.2)
if p_i *are non-uniform* **then**
  $O_\tau \leftarrow$ Solution of Equation (7.1)
$O_n \leftarrow$ Extend O_τ to a Hamiltonian path if $\tau < n$
return O_n

6.1 The best-τ items algorithm

We first consider a case where $p < 1$. From Equation (6.1), we observe that the term p^{n+1} in the numerator diminishes as n increases, and the coefficient W_{O_i} is dominated by $p^{i+1}/(1-p)$. Specifically, when $\lim_{n\to+\infty} p^n = 0$, W_{O_i} *decays exponentially*, meaning the value of $\mathcal{H}(O)$ is dominated by the top items in the order O.

To obtain a better intuition for this regime, consider a scenario where p is very small, i.e., close to 0. In this case, a user is likely to quit after examining only a few items, so the ranking of the first few items becomes critical. To maximize diversity, the most diverse items should be placed at the top, as they contribute most significantly to the objective of the ordered Hamiltonian path.

Building on this insight, we propose an algorithm for the general case where p is bounded away from 1. We call this algorithm *best-τ items* (BτI), where τ is a parameter controlling the approximation quality. The algorithm always outputs a ranking of all items, with the pseudocode provided in Algorithm 1.

Concretely, the best-τ items (BτI) algorithm works as follows: for any integer τ, let

$$\hat{\mathcal{H}}(O_\tau) = \sum_{i=1}^{\tau-1} \frac{p^{i+1}}{1-p} d(\pi(i), \pi(i+1))$$

denote the contribution of O_τ to $\mathcal{H}(O)$. The BτI algorithm seeks the optimal τ-item sequence, denoted as BI_τ, which maximizes $\hat{\mathcal{H}}(O_\tau)$; in other words,

$$\text{BI}_\tau = \underset{O_\tau \sqsubseteq \pi(U)}{\arg\max}\ \hat{\mathcal{H}}(O_\tau). \tag{6.2}$$

The BτI algorithm first construct the ordered sequence of best-τ items, BI_τ, and then extends this sequence to include the remaining items in U. The extension can be done using any arbitrary ordering for the remaining items, as it does not affect the algorithm's approximation ratio. However, a more refined approach, such as a greedy heuristic, could be used to append items that maximize the marginal gain on $\hat{\mathcal{H}}(O_\tau)$ at each step.

The performance of BτI algorithm is as follows.

Theorem 3. *Assume that $p_i = p < 1$, for all $i \in U$. For any integer τ, where $2 \le \tau \le n$, the BτI algorithm obtains a $(1 - p^{\tau-1} - p^{n-\tau} + p^n)$-approximation for MaxOHP.*

Notice that when there is a constant $\epsilon > 0$ such that $p < 1 - \epsilon$, and $\tau \ll n$, both $p^{n-\tau}$ and p^n become negligible. In this case, BτI is a $\left(1 - p^{\tau-1} - \Theta(p^n)\right)$-approximation algorithm. This aligns with our intuition: optimizing the dominant term $\mathcal{H}(O_\tau)$ yields a good approximation for the MaxOHP problem.

Observe that searching the best τ items for BI_τ requires $O(\binom{n}{\tau}\tau!)$ time. Theorem 3 illustrates the inherent trade-off between approximation-quality and efficiency: increasing τ improves the approximation but also increases the computational cost of the algorithm.

Algorithm 2: Greedy matching (GM) algorithm

Input: Finite set U, diversity function d
Output: An ordered sequence of U according to π, $O_n = \pi(U)$

$\tau \leftarrow \lfloor \frac{|U|}{2} \rfloor$, $M \leftarrow$ empty sequence, $S \leftarrow \emptyset$
$E \leftarrow$ all possible pairs (u, v) in decreasing order of $d(u, v)$
for $(u, v) \in E$ **do**
  if $M||(u, v)$ *forms a matching* **then**
    $M \leftarrow M||(u, v)$
    $S \leftarrow S \cup \{u, v\}$
Naming the edges in M:
  $M \leftarrow ((u_1, v_1), (u_3, v_3) \cdots, (u_{2\tau-1}, v_{2\tau-1}))$
if $|U| = 2\tau + 1$ **then**
  Let $v_{2\tau+1} \in U \setminus S$, $\pi(2\tau + 1) \leftarrow v_{2\tau+1}$
for $i \leftarrow \tau$ *to* 1 **do**
  if $i = \tau$ *and* $|U| = 2\tau$ **then**
    $\pi(2\tau - 1) \leftarrow v_{2\tau-1}$, $\pi(2\tau) \leftarrow u_{2\tau-1}$
  if $d(v_{2i-1}, \pi(2i + 1)) \ge d(u_{2i-1}, \pi(2i + 1))$ **then**
    $\pi(2i - 1) \leftarrow u_{2i-1}$, $\pi(2i) \leftarrow v_{2i-1}$
  else
    $\pi(2i - 1) \leftarrow v_{2i-1}$, $\pi(2i) \leftarrow u_{2i-1}$
return $O_n \leftarrow \pi(U)$

6.2 The greedy-matching algorithm

In this section, we examine the case where p is large, specifically when $\lim_{n\to\infty} p = 1$ and $\lim_{n\to\infty} p^n = 0$. This suggests that as the number of items increases, the user accepts more items, but eventually quits before examining all of them. For instance, setting $p = 1 - \frac{1}{\log\log n}$ implies that, in expectation, the user accepts $\log\log n$ items before quitting. For simplicity, we assume $p = 1 - \frac{1}{t_n}$, where t_n is an increasing function of n.

In this scenario, applying BτI algorithm to find best-τ items becomes computationally infeasible. As Theorem 3 shows, achieving a constant approximation requires setting $\tau = \Omega(t_n)$. To address this, we introduce a polynomial-time *greedy-matching* (GM) algorithm.

The idea behind GM is to construct a matching of size-$(t_n/2)$ from t_n items. We observe that for the first t_n items, the coefficient W_{O_i} does not decrease significantly, since $p^{t_n} = 1/e - \Theta(1/t_n)$. Intuitively, the algorithm should find a path of length t_n such that the sum of the edge weights is large. In addition, the algorithm should assign larger W_{O_i} to the edges with larger weights; in other words, the edge weights should be ordered in decreasing order in the path. The proposed GM algorithm, presented in Algorithm 2, achieves both goals.

The GM algorithm starts by applying a greedy algorithm to obtain a matching M, which is a sequence of edges in decreasing order of edge weights. The size of M is equal to $\lfloor \frac{n}{2} \rfloor$. We denote the matching as $M = ((u_{2j-1}, v_{2j-1}))_{j=1}^{\lfloor \frac{n}{2} \rfloor}$.

The next step is to extend M to a Hamiltonian path. Notice that the (undirected) edge (u_{2j-1}, v_{2j-1}) is always the $(2j-1)$-th edge on the path, and the algorithm needs to decide which item comes first in the path. To make this decision, the algorithm utilizes the triangle inequality.

Let π be order of the items on the Hamiltonian path, as obtained by the GM algorithm. The above-mentioned construction satisfies two properties:

(1) $d((\pi(2i-1), \pi(2i))) \geq d((\pi(2i+1), \pi(2i+2)))$, for $i < \lfloor \frac{n}{2} \rfloor$;
(2) $d((\pi(2i), \pi(2i+1))) \geq \frac{1}{2} d((\pi(2i-1), \pi(2i)))$, for $i \leq \lfloor \frac{n}{2} \rfloor$.

For the approximation quality of GM, the following holds.

Theorem 4. *Assume that $p_i = p$, for any $i \in U$. Let p be a function of n such that $\lim_{n\to\infty} p = 1$ and $\lim_{n\to\infty} p^n = 0$. Moreover, assume that $p = 1 - \frac{1}{t_n}$, where t_n is a increasing function of n and $t_n = o(n)$. The algorithm GM yields a $\left(\frac{3(e-1)}{16e^2} - \Theta\left(\frac{1}{t_n}\right)\right)$-approximation for MaxOHP.*

7 Non-uniform continuation probabilities

In this section, we discuss a more general case when the continuation probabilities p_i are non-uniform. We assume $p_i \in [a, b]$, for all $i \in U$, where $0 < a \leq b < 1$. We start by adapting the BτI algorithm, introduced in Section 6, to this setting. Given the computational complexity of the BτI algorithm, we also propose an efficient greedy algorithm, whose performance guarantee is an implication of the BτI algorithm by setting $\tau = 2$.

7.1 The best-τ items algorithm

Recall that the ordered Hamiltonian-path is defined as $\mathcal{H}(O) = \sum_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1))$, where $W_{O_i} = \sum_{j=i+1}^{n} p_{O_j}$. When the continuation probabilities p_i are uniformly equal to p, we observe that the value of W_{O_i} is dominated by $p^{i+1}/(1-p)$, hereby, we can use $p^{i+1}/(1-p)$ to approximate W_{O_i}. However, when p_i are not uniform, it is hard to find a simple and closed-form equation to approximate W_{O_i}, which is convenient for our analysis.

Our solution is to simply truncate W_{O_i}, so that the algorithm can determine the best-τ items by only using the information of the first τ items. In particular, we define

$$\tilde{\mathcal{H}}(O_\tau) = \sum_{i=1}^{\tau-1} \sum_{j=i+1}^{\tau} p_{O_j} d(\pi(i), \pi(i+1)),$$

and we find the best-τ items by maximizing $\tilde{\mathcal{H}}(O_\tau)$. We redefine

$$\mathrm{BI}_\tau = \arg\max_{O_\tau \sqsubseteq \pi(U)} \tilde{\mathcal{H}}(O_\tau). \tag{7.1}$$

The BτI algorithm (Algorithm 1) has the following quality guarantee.

Theorem 5. *Let $p_i \in [a, b]$ for all $i \in U$, where $0 < a \leq b < 1$. Let τ be an integer such that $2 \leq \tau \leq n$. The BτI algorithm provides a $\frac{a^2(1-b)(1-b^{\tau-1})}{a^2+(\tau-1)b^{\tau+1}}$-approximation guarantee for MaxOHP.*

Algorithm 3: Greedy algorithm

Input: Finite set U, probability p_i, diversity function d
Output: An ordered sequence of U according to π, $O_n = \pi(U)$

$O_2 \leftarrow$ Solution of Equation (7.1) by setting $\tau = 2$
for $t = 2;\ t < n;\ t = t + 1$ **do**
  $u \leftarrow \arg\max_{v \in U \setminus O_t} \mathcal{S}_+(O_t || v) - \mathcal{S}_+(O_t)$
  $O_{t+1} \leftarrow O_t || u$
return O_n.

7.2 The greedy algorithm

Since the BτI algorithm is impractical when τ is large, we propose a greedy algorithm, presented in Algorithm 3. Its approximation ratio is obtained by setting $\tau = 2$ in Theorem 5.

Corollary 2. *The greedy algorithm (Algorithm 3) provides a $\frac{a^2(1-b)^2}{a^2+b^2}$-approximation guarantee for MaxOHP.*

8 Experimental evaluation

Our implementation and the datasets are publicly available.[1]

8.1 Datasets

We evaluate our methods on seven public datasets: five commonly used in recommender systems and two, LETOR and LTRC, from the information retrieval field. We refer to these datasets as *recommendation datasets* and *information retrieval datasets*, respectively. Table 1 provides a summary of dataset statistics. Due to space constraints, the dataset URLs are provided in the appendix.

Coat: Coat ratings in the range $[1, 5]$, accompanied by categorical features, such as gender, jacket type, and color.

KuaiRec [14]: Recommendation logs from a video-sharing mobile app, containing each video's duration, categories, and user viewing times. The *watch ratio*, calculated as the user's viewing time divided by the video's duration, and normalized to range [1, 5], serves as personalized rating.

Netflix: Movie ratings in range $[1, 5]$ with genre information. We sample 5 K movies and exclude users with less than 20 interactions.

Movielens: Movie ratings in range $[1, 5]$ with movie genres.

Yahoo: Song ratings in range $[1, 5]$ with song genres. We sample 3 K songs and exclude users with less than 20 interactions.

LETOR: Web-search dataset with relevance scores (in $\{0, 1, 2\}$) on queries-document pairs and document feature vectors.

LTRC: Web-search dataset with relevance scores (in $\{0, 1, 2, 3, 4\}$) on queries-document pairs and document feature vectors.

8.2 Experimental setting

Continuation probability. To better understand how the proposed best-τ algorithm performs across various continuation probabilities, we interpolate these probabilities into four different probability regimes: small ($[0.1, 0.3]$), medium ($[0.4, 0.6]$), large ($[0.7, 0.9]$) and full ($[0.1, 0.9]$). The first three regimes characterize different assumptions about the users' browsing behaviors: users might all

[1] https://github.com/HongLWang/Sequential-diversification-with-provable-guarantees

Table 1: Dataset statistics of the 5 recommendation datasets and 2 information retrieval datasets. $\mathcal{U}$ and $\mathcal{I}$ represent the set of users (or queries) and items (or documents), respectively. $\#\mathcal{R}$ denotes the number of ratings (or relevance scores), $\mathrm{avg}(\mathcal{R})$ denotes the average number of ratings per user (or documents per query), and $\mathrm{avg}(dist)$ denotes the average item-item (or document-document) distance.

Datasets	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\#\mathcal{R}$	$\mathrm{avg}(\mathcal{R})$	$\mathrm{avg}(dist)$
Coat	290	300	6 960	24.0	0.73
KuaiRec	1 411	3 327	4 676 570	3 314.4	0.91
Netflix	4 999	1 112	557 176	1 121.5	0.83
Movielens	6 040	3 706	1 000 208	165.6	0.83
Yahoo	21 181	3 000	963 296	45.8	0.26
LETOR	1 691	65 316	69 614	41.2	0.33
LTRC	5 154	146 995	152 772	29.60	0.26

be highly likely to accept the system's recommendations or tend to easily quit the system or in the middle. The last probability regime does not impose any specific assumptions.

We obtain the continuation probabilities in different ways for the recommendation datasets and the information-retrieval datasets. For the former, we first complete the user-item rating matrix using a matrix-factorization approach [23]. The estimated ratings, which range from $[1, 5]$, are then interpolated into the four probability regimes. For the latter datasets, we treat each query and its corresponding documents as a dataset and directly interpolate the relevance scores into probability regimes. This is meaningful as the documents of different queries have almost no intersection.

Distance function. For the recommendation datasets, given the categorical nature of the data, we use the *Jaccard distance* as the item-item distance function. In particular, if C_i denotes the set of categories of item i, the Jaccard distance $d(i, j)$ is defined as $d(i, j) = 1 - |C_i \cap C_j|/|C_i \cup C_j|$. For the information-retrieval datasets, with the provided document feature vectors, we use the 1−*cosine distance* as the document-document distance function.

Additional metrics. To evaluate the performance of our algorithms in terms of user engagement and satisfaction, we use two metrics: expected discounted cumulative gain (*ExpDCG*) and expected serendipity (*ExpSerendipity*). Discounted cumulative gain [21] measures the accumulated relevance of items in a ranking list, discounted by their position. Serendipity [18] quantifies how relevant and unexpected the recommended items are to the user. We adapt these metrics to account for users' continuation probabilities. For any user, let p_i be the continuation probability of item i.

ExpDCG is defined as $\sum_{j=1}^{|\mathcal{I}|} \left(\sum_{t=1}^{j} \frac{p_t}{log_2(t+1)} \right) (1 - p_{j+1}) \prod_{t=1}^{j} p_t$. For serendipity, let H be the set of previously rated items, and $C(H)$ be the set of categories covered by H. *ExpSerendipity* is defined as $\sum_{j=1}^{|\mathcal{I}|} \left(\sum_{t=1}^{j} p_t \cdot I(t, u) \right) (1 - p_{j+1}) \prod_{t=1}^{j} p_t$, where $I(t, u) = 1$ if $|C(H \cup \{t\})| > |C(H)|$ and 0 otherwise. Here, p_t assesses the relevance of item t, while $I(t, u)$ captures its unexpectedness.

Note that, by definition, the *ExpSerendipity* score for the LETOR and LTRC datasets is 0 as the sequences for each user comprise only rated items. Similarly, the KuaiRec dataset also has a *ExpSerendipity* score of 0, as each user has already covered all categories in their history of ratings. Therefore, we omit the *ExpSerendipity* scores for these datasets.

8.3 Proposed methods and baselines

First, we present the baseline methods used in our experiments. We reformulate the baselines using our notation to enhance clarity. We consistently denote by R the set of already-selected items. The parameter λ models the trade-off diversity vs. relevance. We conduct a grid search to find $\lambda \in [0, 1]$ that performs the best according to the MaxSSD objective for baselines MSD, MMR, and DPP. We report the results using that optimal value of λ for each method.

Random. Randomly shuffle all items into a sequence.

EXPLORE [11]. Details on the EXPLORE algorithm and its adaptation to our setting are provided in the appendix.

Diversity-weighted utility maximization (DUM) [4]. DUM seeks to find a permutation π that maximizes a diversity-weighted relevance objective: $\sum_{i=1}^{|U|} (|C(O_i)| - |C(O_{i-1})|)\, p_{\pi(i)}$, where $C(O_i)$ denotes the set of categories that O_i covers.

Maximal marginal relevance (MMR) [6]. MMR iteratively selects the item $i \in U$ that maximizes $\lambda\, p_i - (1 - \lambda) \max_{j \in R} (1 - d(i, j))$.

Max-sum diversification (MSD) [5]. MSD iteratively selects the item $i \in U$ that maximizes $p_i + \lambda \sum_{j \in R} d(i, j)$.

Determinantal point process (DPP) [10]. DPP iteratively selects the item i that maximizes $\lambda p_i + (1 - \lambda)(\log\det(\mathrm{S}_{R \cup \{i\}}) - \log\det(\mathrm{S}_R))$, where S is a similarity matrix, with $\mathrm{S}_{ij} = 1 - d(i, j)$.

Next, we provide more details of our methods BτI and BτI-H.

BτI. We refer to the BτI algorithm as B2I, B3I, and B4I when τ is set to be 2, 3, and 4. Once the first τ-sequence is chosen, we greedily extend the remaining items that maximize the incremental gain of MaxSSD. Notice that B2I in our implementation is the same as the greedy algorithm we propose in Algorithm 3.

BτI-H. Since BτI can be computationally expensive when τ is large, we adopt a greedy approach by selecting the top 100 items that maximize the incremental gain of MaxSSD, thereby forming a candidate set. The best τ items are exclusively selected within this candidate set. We extend the best τ items in the same way as in BτI.

Throughout the experiments, we evaluate B2I, B3I, and B4I on dataset LETOR and LTRC. For the recommendation datasets, we evaluate B2I and heuristics B3I-H and B4I-H on these datasets.

8.4 Results and discussion

Performance on sequential sum diversity. We conduct experiments on all datasets evaluating our methods and the baselines. For each dataset and method, we obtain a sequence O of all items for each user and calculate the sequential sum diversity $\mathcal{S}_+(O)$. We report the average $\mathcal{S}_+(O)$ values and standard deviations across all users as shown in Table 2. Results for the other regimes are included in the appedix.

In Table 2, we observe that B2I achieves excellent results and outperforms all baselines. The only exception is on KuaiRec, where B4I-H marginally outperforms B2I and achieves the best result.

Next, compared with B2I, the performance of B3I (or B3I-H) and B4I (or B4I-H) declines. This decline can be explained by two factors. First, the BτI and BτI-H algorithms select the top τ items by solving the intermediate MaxOHP problem, which leads to some information loss compared to directly solving MaxSSD, as their objectives differ when $\tau > 2$. Second, due to the relatively low continuation

Table 2: Sequential sum diversity $\mathcal{S}_+(O)$ with item continuation probability mapped to $[0.4, 0.6]$. The results marked with * are obtained using the BτI-H heuristic.

Datasets	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
Coat	0.646 ± 0.177	1.276 ± 0.374	1.214 ± 0.348	1.211 ± 0.349	1.282 ± 0.377	<u>1.284 ± 0.377</u>	**1.289 ± 0.376**	1.184 ± 0.368*	1.184 ± 0.367*
KuaiRec	0.383± 0.066	0.681 ± 0.082	0.411 ± 0.086	0.704 ± 0.057	0.725 ± 0.080	0.711 ± 0.073	<u>0.778 ± 0.075</u>	0.776 ± 0.072*	**0.782 ± 0.073***
Netflix	0.921± 0.220	1.861 ± 0.263	1.779 ± 0.241	1.888 ± 0.254	1.937 ± 0.258	1.933 ± 0.258	**1.946 ± 0.264**	1.919 ± 0.270*	1.910 ± 0.272*
Movielens	0.878± 0.246	2.142 ± 0.114	2.080 ± 0.113	2.228 ± 0.046	2.232 ± 0.044	2.230 ± 0.046	**2.235 ± 0.043**	<u>2.234 ± 0.046</u>*	2.233 ± 0.050*
Yahoo	0.231 ± 0.248	1.684 ± 0.465	<u>1.793 ± 0.466</u>	<u>1.793 ± 0.466</u>	<u>1.793 ± 0.466</u>	<u>1.793 ± 0.466</u>	**1.795 ± 0.466**	1.729 ± 0.495*	1.715 ± 0.496*
LETOR	0.208 ± 0.106	0.440 ± 0.190	0.564 ± 0.184	0.577 ± 0.191	0.570 ± 0.191	0.557 ± 0.181	**0.592 ± 0.195**	0.582 ± 0.197	<u>0.588 ± 0.197</u>
LTRC	0.359 ± 0.116	0.597 ± 0.151	0.640 ± 0.134	0.662 ± 0.146	0.644 ± 0.145	0.642 ± 0.138	**0.670 ± 0.144**	0.667 ± 0.145	<u>0.668 ± 0.145</u>

Table 3: ***ExpDCG*** **and** ***ExpSerendipity*** **values with item continuation probabilities mapped to $[0.4, 0.6]$. The results marked with * are obtained using the BτI-H heuristic.**

Metrics	Datasets	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
ExpDCG	Coat	0.354 ± 0.070	0.496±0.100	**0.513 ± 0.096**	0.489 ± 0.091	<u>0.511 ± 0.095</u>	0.510 ± 0.095	0.494 ± 0.096	0.499 ± 0.096*	0.499 ± 0.096*
	KuaiRec	0.347 ± 0.049	0.483 ± 0.048	**0.590 ± 0.048**	0.459 ± 0.028	0.465 ± 0.030	0.472 ± 0.032	0.409 ± 0.072	0.477 ± 0.031*	<u>0.480 ± 0.032</u>*
	Netflix	0.413 ± 0.065	0.575 ± 0.055	**0.604 ± 0.044**	0.596 ± 0.044	<u>0.600 ± 0.044</u>	<u>0.600 ± 0.044</u>	0.597 ± 0.046	<u>0.600 ± 0.044</u>*	<u>0.600 ± 0.044</u>*
	Movielens	0.394 ± 0.080	0.620 ± 0.026	**0.635 ± 0.005**	**0.635 ± 0.005**	**0.635 ± 0.005**	**0.635 ± 0.005**	**0.635 ± 0.005**	**0.635 ± 0.005***	**0.635 ± 0.005***
	Yahoo	0.360 ± 0.094	0.550 ± 0.100	<u>0.572 ± 0.086</u>	<u>0.572 ± 0.086</u>	<u>0.572 ± 0.086</u>	<u>0.572 ± 0.086</u>	0.565 ± 0.092	**0.577 ± 0.085***	0.577 ± 0.085*
	LETOR	0.274 ± 0.082	0.330±0.104	0.327 ± 0.104	**0.416 ± 0.123**	0.400 ± 0.115	<u>0.406 ± 0.117</u>	0.326 ± 0.104	0.390 ± 0.108	0.393 ± 0.110
	LTRC	0.323 ± 0.068	0.373±0.082	0.378 ± 0.078	<u>0.443 ± 0.084</u>	**0.447 ± 0.088**	0.440 ± 0.081	0.378 ± 0.078	0.402 ± 0.080	0.407 ± 0.080
ExpSerendipity	Coat	0.141 ± 0.138	0.247±0.186	0.204 ± 0.159	0.225 ± 0.147	0.214 ± 0.157	0.218 ± 0.160	0.288 ± 0.202	**0.354 ± 0.238***	<u>0.351 ± 0.240</u>*
	Netflix	0.028 ± 0.068	0.132±0.151	0.133 ± 0.171	**0.152 ± 0.164**	0.148 ± 0.164	<u>0.149 ± 0.163</u>	0.143 ± 0.161	0.139 ± 0.162*	0.139 ± 0.165*
	Movielens	0.043 ± 0.091	**0.141±0.153**	<u>0.136 ± 0.159</u>	0.115 ± 0.144	0.115 ± 0.144	0.115 ± 0.144	0.115 ± 0.144	0.115 ± 0.144*	0.115 ± 0.144*
	Yahoo	0.040 ± 0.079	0.275±0.145	<u>0.309 ± 0.140</u>	0.303 ± 0.136	0.303 ± 0.136	0.303 ± 0.136	**0.325 ± 0.143**	0.264 ± 0.143*	0.259 ± 0.145*

probabilities, the expected number of items a user accepts before quitting is typically not larger than 2. Consequently, B2I, which focuses on optimizing the top 2 items, tends to perform better than B3I or B4I, which may produce suboptimal top-2 sequences.

Next, when comparing the performance of B3I and B4I, we observe that performance increases as τ increases. However, when comparing the performance of B3I-H and B4I-H, no consistent trend in the algorithms' performance is observed. This inconsistency can be attributed to the filtering process applied to these heuristics. It is worth noting that similar observations can be made across all probability regimes, as detailed in the appendix.

When comparing the performances of BτI and BτI-H across different continuation probabilities, we observe both algorithms perform better with smaller continuation probabilities than with larger ones. This observation aligns with our theoretical analysis on approximation guarantees (Theorem 5). With fixed τ, the approximation ratio decreases as the continuation probability increases.

Performance on ***ExpDCG*** **and** ***ExpSerendipity*****.** With respect to the *ExpDCG* metric, we observe in Table 3 that DUM has the best performance, achieving the highest or second-highest scores in 5 out of 7 datasets. This result is consistent with the theory, as the algorithm prioritizes selecting items with the highest relevance scores. The BτI and BτI-H algorithms do not achieve the highest scores, which is expected as they do not prioritize optimizing relevance. It is worth noting, however, that the differences across the methods are relatively small. Similar findings hold across all the probability regimes.

Regarding the *ExpSerendipity* metric, we observe no clear winners among all methods (with MMR performing poorly), as each method achieves (at least) second-highest scores in 1 out of 4 datasets (Table 3). However, when considering all the continuation probability regimes, our approaches are better. This suggests that the BτI (-H) algorithm may have a higher potential for generating more exploratory recommendations.

9 Conclusion

In this paper, we study the novel concept of *sequential diversity*, aiming to instill diversity into rankings while considering the relevance of items and modeling user behavior. The framework is designed to find rankings that consist of items that are both diverse and relevant. The formulation gives rise to a novel computational problem, for which we establish a connection with the ordered Hamiltonian-path problem and design approximation algorithms with provable guarantees. Our algorithms offer trade-offs of efficiency vs. approximation quality.

Our paper opens many exciting directions for future work. First, it will be interesting to devise more efficient combinatorial algorithms without losing their approximation guarantees. Additionally, it will be valuable to incorporate user models of higher complexity into the framework, in order to capture more nuanced user behavior. Last, it will be valuable to study the proposed framework with a user study on a real-world system.

Acknowledgments

We sincerely thank Ilie Sarpe for carefully reviewing our manuscript and providing timely, detailed feedback, which helps to improve the quality of this paper. This research is supported by the ERC Advanced Grant REBOUND (834862), the EC H2020 RIA project SoBigData++ (871042), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] Mustafa Abdool, Malay Haldar, Prashant Ramanathan, Tyler Sax, Lanbo Zhang, Aamir Manaswala, Lynn Yang, Bradley Turnbull, Qing Zhang, and Thomas Legrand. 2020. Managing diversity in airbnb search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2952–2960.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining*. 5–14.

[3] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2014. Diversified Utility Maximization for Recommendations.. In *RecSys Posters*.

[4] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal Greedy Diversity for Recommendation. In *IJCAI*, Vol. 15. 1742–1748.

[5] Allan Borodin, Hyun Chul Lee, and Yuli Ye. 2012. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*. 155–166.

[6] Jaime Carbonell and Jade Goldstein. 1998. 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*. 335–336.

[7] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. 2015. Max-sum diversity via convex programming. *arXiv preprint arXiv:1511.07077* (2015).

[8] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. 2017. Local search for max-sum diversification. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 130–142.

[9] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. 2019. An improved analysis of local search for max-sum diversification. *Mathematics of Operations Research* 44, 4 (2019), 1494–1509.

[10] Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems* 31 (2018).

[11] Erica Coppolillo, Giuseppe Manco, and Aristides Gionis. 2024. Relevance Meets Diversity: A User-Centric Framework for Knowledge Exploration Through Recommendations. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 490–501. https://doi.org/10.1145/3637528.3671949

[12] Zdravko Cvetkovski. 2012. *Inequalities: theorems, techniques and selected problems*. Springer Science & Business Media.

[13] Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. 2020. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2001–2004.

[14] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 540–550.

[15] Michael R Garey and David S Johnson. 1979. Computers and intractability. *A Guide to the* (1979).

[16] Sreenivas Gollapudi and Aneesh Sharma. 2009. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*. 381–390.

[17] Yuri Gurevich and Saharon Shelah. 1987. Expected computation time for Hamiltonian path problem. *SIAM J. Comput.* 16, 3 (1987), 486–502.

[18] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[19] Yanhua Huang, Weikun Wang, Lei Zhang, and Ruiwen Xu. 2021. Sliding spectrum decomposition for diversified recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3041–3049.

[20] Samuel Ieong, Mohammad Mahdian, and Sergei Vassilvitskii. 2014. Advertising in a stream. In *Proceedings of the 23rd International Conference on World Wide Web* (Seoul, Korea) *(WWW '14)*. Association for Computing Machinery, New York, NY, USA, 29–38. https://doi.org/10.1145/2566486.2568030

[21] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[22] Jon Kleinberg, Emily Ryu, and Éva Tardos. 2023. Calibrated recommendations for users with decaying attention. *arXiv preprint arXiv:2302.03239* (2023).

[23] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[24] Shameem Puthiya, Nicolas Usunier, and Yves Grandvalet. 2016. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 15–22.

[25] Shaojie Tang, Jing Yuan, and Vijay Mookerjee. 2020. Optimizing ad allocation in mobile advertising. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 181–190.

[26] Haolun Wu, Yansen Zhang, Chen Ma, Fuyuan Lyu, Bowei He, Bhaskar Mitra, and Xue Liu. 2024. Result Diversification in Search and Recommendation: A Survey. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[27] Yue Xu, Hao Chen, Zefan Wang, Jianwen Yin, Qijie Shen, Dimin Wang, Feiran Huang, Lixiang Lai, Tao Zhuang, Junfeng Ge, et al. 2023. Multi-factor sequential re-ranking with perception-aware diversification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5327–5337.

[28] ChengXiang Zhai, William W Cohen, and John Lafferty. 2015. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Acm sigir forum*, Vol. 49. ACM New York, NY, USA, 2–9.

[29] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

10 ETHICS STATEMENT

In this work, we introduce BτI, a method designed to generate rankings of items that are both relevant and diverse for users. The proposed BτI and BτI-H algorithms can enhance the user experience by preventing the so-called "rabbit hole" effect, where users may become trapped in a narrow set of recommendations or search results that lack variety. By maximizing the sequential diversity of the top-ranked items in a user's recommendation list or web search results, our approach encourages exploration and a broader discovery of content, thereby reducing the risk of reinforcing cognitive biases or limiting exposure to diverse perspectives.

To ensure the ethical integrity of our work, all datasets utilized in this study are publicly available and have been processed to protect user privacy. Specifically, user IDs have been anonymized and replaced with numeric identifiers, ensuring that no personal or identifying information is disclosed.

11 Omitted proofs from Section 5

Lemma 1. *The sequential sum diversity objective in Equation (3.1) can be reformulated as*

$$\mathcal{S}_+(O) = \sum\nolimits_{i=1}^{n-1} p_{O_{i+1}} d(\pi(i+1), O_i).$$

Proof. We start with demonstrating that the sequential sum-diversity objective can be precisely reformulated via the expression $2\sum_{i=2}^{n}\sum_{j=1}^{i-1}\prod_{t=1}^{i}\Pr_{\pi(t)} d(\pi(i),\pi(j))$. Next, the formulation can be simplified using the definition of p_{O_i} and $d(\pi(i), O_{i-1})$.

We write the definition of sequential sum-diversity objective, and proceed by merging the terms according to our observations.

$$\begin{aligned}
&\sum_{k=1}^{n}\Pr(A=O_k)\sum_{\pi(i),\pi(j)\in O_k} d(\pi(i),\pi(j))\\
&\overset{(a)}{=}\sum_{k=1}^{n}\Pr(A=O_k)\sum_{i,j\le k} d(\pi(i),\pi(j))\\
&\overset{(b)}{=}\sum_{k=1}^{n-1}\Big(\prod_{t=1}^{k}p_{\pi(t)}-\prod_{t=1}^{k+1}p_{\pi(t)}\Big)\sum_{i,j\le k} d(\pi(i),\pi(j))\\
&+\prod_{t=1}^{n}p_{\pi(t)}\sum_{i,j\le n} d(\pi(i),\pi(j))\\
&=\sum_{k=1}^{n-1}\prod_{t=1}^{k+1}p_{\pi(t)}\Big(-\sum_{i,j\le k} d(\pi(i),\pi(j))+\sum_{i,j\le k+1} d(\pi(i),\pi(j))\Big)\\
&=\sum_{k=1}^{n-1}\prod_{t=1}^{k+1}p_{\pi(t)}\Big(\sum_{i=k+1,j\le k} d(\pi(i),\pi(j))+\sum_{j=k+1,i\le k} d(\pi(i),\pi(j))\Big)\\
&=2\sum_{k=1}^{n-1}\prod_{t=1}^{k+1}p_{\pi(t)}\Big(\sum_{i=k+1,j\le k} d(\pi(i),\pi(j))\Big)\\
&=2\sum_{k=2}^{n}\prod_{t=1}^{k}p_{\pi(t)}\Big(\sum_{i=k,j\le k-1} d(\pi(i),\pi(j))\Big)\\
&=2\sum_{i=2}^{n}\sum_{j=1}^{i-1}\prod_{t=1}^{i}p_{\pi(t)} d(\pi(i),\pi(j)).
\end{aligned} \tag{11.1}$$

Notice that equality (a) holds by the definition of $\pi(i)$: indeed, $\pi(i)$ indicates the item that is placed at the i-th position. Equality (b) holds by substituting the expression of $\Pr(A = O_k)$ from the observation 1. For the following equations, we re-arrange the simplify the formulas. □

For the proof of Observation 2 and Observation 3, we use two lemma from Kleinberg et al. [22].

Lemma 2 (Kleinberg et al. [22]). *If f and g are ordered-submodular, then $\alpha f + \beta g$ is also ordered-submodular for any $\alpha, \beta \geq 0$.*

Lemma 3 (Kleinberg et al. [22]). *Suppose h is a monotone submodular set function, Then the function f constructed by evaluating h on the set of the first t elements of S, that is,*

$$f(S) = \begin{cases} h(S) & \text{if } |S| \le t \\ h(S_t) & \text{if } |S| > t \end{cases}$$

is ordered-submodular.

Observation 2. *The sequential coverage diversity function $\mathcal{S}_c(\cdot)$ is ordered-submodular.*

Proof. By the definition of the $\mathcal{S}_c$ objective we have:

$$\mathcal{S}_c(O) = \mathbb{E}_{A\sqsubseteq O}[\mathcal{D}(A)] = \sum_{k=1}^{n}\Pr(A = O_k)\mathcal{D}(O_k), \tag{11.2}$$

where $\mathcal{D}(O_k)$ is a monotone submodular function. Let $f_k(O) = \mathcal{D}(O_k)$ so we can rewrite $\mathcal{S}_c(O)$ as $\mathcal{S}_c(O) = \sum_{k=1}^{n} p_{O_k} f_k(O)$. Since $f_k(O)$ is a monotone submodular function evaluated on the first k items, by Lemma 3, $f_k(O)$ is an ordered submodular function. Since $\Pr(O_k) \geq 0$ for all k, by Lemma 2, $\mathcal{S}_c(\cdot)$ is ordered-submodular. □

Observation 3. *The sequential sum diversity function $\mathcal{S}_+(\cdot)$ is not ordered-submodular.*

Proof. To show $\mathcal{S}_+(\cdot)$ is not ordered submodular, we only need one counter example. Let X and Y be two sequences that satisfy $p_X > 0$ and $p_Y > 0$. Let s be an item such that $p_s > 0$ and $d(s, Y) > 0$. Also, let $\bar{s}$ be an item such that $p_{\bar{s}} = 0$. It follows that

$$\begin{aligned}
\mathcal{S}_+(X||s||Y) - \mathcal{S}_+(X||\bar{s}||Y) &= \mathcal{S}_+(X||s||Y) - \mathcal{S}_+(X)\\
&> \mathcal{S}_+(X||s) - \mathcal{S}_+(X).
\end{aligned}$$

This example violates the definition of ordered submodularity, which proves that $\mathcal{S}_+(\cdot)$ is not an ordered submodular function. □

Theorem 1. *MaxSSD is* **NP***-hard, even when all p_i are equal.*

Proof. We provide a reduction from the decision version of the maximum clique problem to the decision version of the MaxSSD problem. Define CLIQUE $= (G, V, k)$, with $|V| = n$ to describe instances of the maximum clique problem, where we want to decide whether there exists a clique of size k in G. Also define an instance of the MaxSSD problem by (U, d, p, θ), where U is the item set, p is a uniform continuation probability, and $d(\cdot,\cdot)$ is a metric distance function, and we want to decide whether there exist an ordering π of U such that $\mathcal{S}_+(\pi(U)) > \theta$.

Given an instance of CLIQUE, we construct an instance of MaxSSD in the following way: we set $U = V$, and we assign $d(u, v) = 2$ if there is an edge between nodes u and v in G, otherwise we set $d(u, v) = 1 + \epsilon$, where ϵ is any constant smaller than 1. We set $p = \frac{1-\epsilon}{2n^2}$ and $\theta = 2\sum_{i=1}^{k-1} i \times \left(\frac{1-\epsilon}{2n^2}\right)^{i+1}$. One can easily verify that this transformation can be done in polynomial time, and that $d(\cdot,\cdot)$ is a metric.

If there is a clique of size $k \le n$ in G, then for the corresponding item set $U = V$, we can construct an ordered sequence $O = (\pi(i))_{i=1}^{n}$ by assigning O_k to be the k nodes in U that corresponds to the clique in G, and setting in an arbitrary way the rest of the sequence order.

Let $\mathcal{L}(O)$ be a lower bound of $\mathcal{S}_+(A)$. One can verify it is also a lower bound for the optimal objective value of the MaxSSD problem. We can define $\mathcal{L}(O)$ as follows:

$$\begin{aligned}\mathcal{L}(O) &= \sum_{i=1}^{k-1} p_{O_{i+1}} d(\pi(i+1), O_i) \\ &= 2(p^2 + 2p^3 + 3p^4 + \cdots + (k-2)p^{k-1} + (k-1)p^k) \\ &= 2\sum_{i=1}^{k-1} i \times p^{i+1}.\end{aligned}$$

Furthermore, it holds that $\mathcal{L}(O) = \theta$.

On the other hand, if there does not exist a size k clique in G, then the densest possible graph structure of G contains $T = \lfloor \frac{n}{k-1} \rfloor$ size-$(k-1)$ cliques, and one clique of size $L = n - (k-1)T$. A node in a size-$(k-1)$ clique can connect to at most $k-2$ nodes in another size-$(k-1)$ clique, thus the maximum number of edges between any two size-$(k-1)$ cliques is $(k-1)(k-2)$. Similarly, the number of edges between a size-L clique and a size-$(k-1)$ clique is $(k-2)L$. Given this graph structure, we can construct an ordered sequence $\tilde{O} = (\tilde{\pi}(i))_{i=1}^n$, where sequence $[\tilde{\pi}((j-1)(k-1)+1), \cdots, \tilde{\pi}((k-1)\times j)]$ corresponds to the j-th size-$(k-1)$ clique of G, with $j \in [1, T]$, and the remaining sequence corresponds to the size-L clique of G.

Let $\mathcal{U}(\tilde{O}) = \mathcal{S}_+(\tilde{O})$. One can verify that $\mathcal{U}(\tilde{O})$ is an upper bound of the optimal objective value of the corresponding MaxSSD problem. It holds that

$$\begin{aligned}\mathcal{U}(\tilde{O}) &= \sum_{i=1}^{n-1} p_{\tilde{O}_{i+1}} d(\tilde{\pi}(i+1), \tilde{O}_i) \\ &\overset{(a)}{=} 2(p^2 + 2p^3 + \cdots + (k-2)p^{k-1}) \\ &+ \sum_{i=1}^{T-1}\sum_{j=0}^{k-2} p^{j+1+i(k-1)}(2[i(k-2)+j] + i(1+\epsilon)) \\ &+ \sum_{j=0}^{L-1} p^{T(k-1)+j+1}(2[T(k-2)+j] + T(1+\epsilon)) \\ &\overset{(b)}{<} \mathcal{L}(O) + (\epsilon-1)p^k + 2n(p^{k+1} + p^{k+2} + \cdots + p^n),\end{aligned} \tag{11.3}$$

where (a) holds because of Lemma 1 and (b) holds because for $j \in [k+1, n]$, the j-th node contributes at most $2np^j$ to $\mathcal{U}(\tilde{O})$.

Furthermore, since

$$\begin{aligned}\mathcal{L}(O) - \mathcal{U}(\tilde{O}) &= \theta - \mathcal{U}(\tilde{O}) \\ &> (1-\epsilon)p^k - 2n(p^{k+1} + p^{k+2} + \cdots + p^n) \\ &= (1-\epsilon)p^k - 2np(p^k + p^{k+1} + \cdots + p^{n-1}) \\ &> (1-\epsilon)p^k - 2np(n-k)p^k \\ &= p^k(1-\epsilon-2np(n-k)) = \frac{p^k(1-\epsilon)k}{n} > 0\end{aligned}$$

holds for all k, we conclude that a yes instance of the CLIQUE problem indicates a yes instance of the MaxSSD problem, and a no instance indicates a no instance of the MaxSSD problem.

This finishes our reduction from the CLIQUE problem to the MaxSSD problem and we conclude that the MaxSSD problem is **NP**-hard. □

11.1 Proof of Theorem 2

We start by observing that $\mathcal{H}(O)$ can be reformulated according to Lemma 4. The reformulation helps us to relate $\mathcal{H}(O)$ with $\mathcal{S}_+(O)$. Notice that $p_{O_{i+1}}$ is the common coefficient of $d_L(O_{i+1})$ in $\mathcal{H}(O)$ and $d(\pi(i+1), O_i)$ in $\mathcal{S}_+(O)$. Hence, we essentially only compare $d_L(O_{i+1})$ with $d(\pi(i+1), O_i)$, for all $1 \le i \le n-1$.

Lemma 4. *$\mathcal{H}(O)$ can be equivalently formulated as*

$$\mathcal{H}(O) = \sum\nolimits_{i=1}^{n-1} p_{O_{i+1}} d_L(O_{i+1}),$$

where $d_L(O_k) \coloneqq \sum_{t=1}^{k-1} d(\pi(t), \pi(t+1))$.

The following corollary is a direct implication from the triangle inequality, $2d(\pi(i+1), O_i) \ge d_L(O_{i+1})$, for any sequence O.

Corollary 3. *For any ordered sequence O of items in U it holds*

$$2\mathcal{S}_+(O) \ge \mathcal{H}(O).$$

Let O^* be the optimal sequence of $\mathcal{S}_+$: $O^* = \arg\max \mathcal{S}_+(O)$. Let $(\pi(i), \pi(i+1))$ be any pair of items swapped in O^*, and denote $O' = (\ldots, \pi(i+1), \pi(i), \ldots)$. By optimality, it follows that $\mathcal{S}_+(O^*) \ge \mathcal{S}_+(O')$. With some simplification, we get the following lemma.

Lemma 5. *For an optimal solution $O^* = \arg\max \mathcal{S}_+(O)$ to the MaxSSD problem, it holds*

$$d(\pi(i+1), O_i^*) \le \frac{1 - p_{\pi(i+1)}}{p_{\pi(i+1)}} \sum\nolimits_{j=1}^{i} \frac{p_{\pi(j+1)}}{1 - p_{\pi(j+1)}} d(\pi(j), \pi(j+1)),$$

for any $1 \le i \le n-1$.

We observe that $\frac{1-p_i}{p_i} \le \frac{1-a}{a}$, and $\frac{p_i}{1-p_i} \le \frac{b}{1-b}$ if $p_i \in [a, b]$. The following corollary holds directly by the implication of Lemma 5 that $d(\pi(i+1), O_i^*) \le \frac{b(1-a)}{a(1-b)} d_L(O_{i+1}^*)$.

Corollary 4. *For an optimal sequence $O^* = \arg\max \mathcal{S}_+(O)$ to the MaxSSD problem, and assuming that $p_i \in [a, b]$, for all $i \in U$, it holds that $\mathcal{S}_+(O^*) \le \frac{b(1-a)}{a(1-b)} \mathcal{H}(O^*)$.*

We are ready to prove Theorem 2.

Proof. Let O^* be an optimal sequence for the MaxSSD problem, and O^o be an optimal sequence for the MaxOHP problem. Let O^α be an α-approximation sequence for MaxOHP. It holds

$$\begin{aligned}\mathcal{S}_+(O^*) &\overset{(\dagger)}{\le} \frac{b(1-a)}{a(1-b)} \mathcal{H}(O^*) \le \frac{b(1-a)}{a(1-b)} \mathcal{H}(O^o) \\ &\le \frac{b(1-a)}{a(1-b)} \frac{\mathcal{H}(O^\alpha)}{\alpha} \overset{(\ddagger)}{\le} \frac{2b(1-a)}{\alpha a(1-b)} \mathcal{S}_+(O^\alpha),\end{aligned}$$

where inequality $(\dagger)$ follows from Corollary 4, and $(\ddagger)$ from Corollary 3. Thus we proved $\frac{\mathcal{S}_+(O^\alpha)}{\mathcal{S}_+(O^*)} \ge \frac{a(1-b)}{2b(1-a)} \alpha$. □

Lemma 4. *$\mathcal{H}(O)$ can be equivalently formulated as*

$$\mathcal{H}(O) = \sum\nolimits_{i=1}^{n-1} p_{O_{i+1}} d_L(O_{i+1}),$$

where $d_L(O_k) \coloneqq \sum_{t=1}^{k-1} d(\pi(t), \pi(t+1))$.

Proof. We simply re-arrange the formulas.

$$\begin{aligned}\mathcal{H}(O) &= \sum_{i=1}^{n-1} w_i d(\pi(i), \pi(i+1)) \\ &= \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} p_{O_k} d(\pi(i), \pi(i+1)) \\ &= p_{O_2} d(\pi(1), \pi(2)) + p_{O_3}(d(\pi(1), \pi(2)) + d(\pi(2), \pi(3))) \\ &+ \cdots + p(O) \sum_{t=1}^{n-1} d(\pi(t), \pi(t+1)) \\ &= p_{O_2} d_L(O_2) + p_{O_3} d_L(O_3) + \cdots + p(O) d_L(O) \\ &= \sum_{i=1}^{n-1} p_{O_{i+1}} d_L(O_{i+1}).\end{aligned} \tag{11.4}$$

□

Corollary 3. *For any ordered sequence O of items in U it holds*

$$2\mathcal{S}_+(O) \geq \mathcal{H}(O).$$

Proof. First, we show that if $2d(\pi(i+1), O_i) \geq d_L(O_{i+1})$ holds for any $i \in [n-1]$, then our lemma is proven:

$$\begin{aligned}2\mathcal{S}_+(O) &= 2\sum_{i=1}^{n-1} p_{O_{i+1}} d(\pi(i+1), O_i) \\ &= \sum_{i=1}^{n-1} p_{O_{i+1}} 2d(\pi(i+1), O_i) \\ &\geq \sum_{i=1}^{n-1} p_{O_{i+1}} d_L(O_{i+1}) \\ &= \mathcal{H}(O).\end{aligned}$$

Next, we show that $2d(\pi(i+1), O_i) \geq d_L(O_{i+1})$ holds for any $1 \leq i \leq n-1$:

$$\begin{aligned}d_L(O_{i+1}) &= \sum_{j=1}^{i} d(\pi(j), \pi(j+1)) \\ &\overset{(a)}{\leq} \sum_{j=1}^{i} [d(\pi(i+1), \pi(j)) + d(\pi(i+1), \pi(j+1))] \\ &= \sum_{j=1}^{i} d(\pi(i+1), \pi(j)) + \sum_{j=1}^{i} d(\pi(i+1), \pi(j+1)) \\ &= d(\pi(i+1), O_i) + d(\pi(i+1), O_{i+1}) - d(\pi(i+1), \pi(1)) \\ &\overset{(b)}{=} d(\pi(i+1), O_i) + d(\pi(i+1), O_i) - d(\pi(i+1), \pi(1)) \\ &< 2d(\pi(i+1), O_i),\end{aligned}$$

where (a) holds by triangle inequality, and (b) holds since $d(\pi(i+1), \pi(i+1)) = 0$. □

Lemma 5. *For an optimal solution $O^* = \arg\max \mathcal{S}_+(O)$ to the Max-SSD problem, it holds*

$$d(\pi(i+1), O_i^*) \leq \frac{1 - p_{\pi(i+1)}}{p_{\pi(i+1)}} \sum\nolimits_{j=1}^{i} \frac{p_{\pi(j+1)}}{1 - p_{\pi(j+1)}} d(\pi(j), \pi(j+1)),$$

for any $1 \leq i \leq n-1$.

Proof. Let $O^* = (\pi(i))_{i=1}^n$ be an optimal sequence, and let us denote $O' = (\ldots, \pi(i+1), \pi(i), \ldots)$ to be the sequence after swapping node pair $\{\pi(i), \pi(i+1)\}$. Then,

$$\begin{aligned}\mathcal{S}_+(O') &= \sum_{j=1}^{i-2} p_{O^*_{j+1}} d(\pi(j+1), O^*_j) + p_{\pi(i+1)} p_{O^*_{i-1}} d(\pi(i+1), O^*_{i-1}) \\ &+ p_{\pi(i)} p_{\pi(i+1)} p_{O^*_{i-1}} d(\pi(i), O^*_{i-1} \cup \pi(i+1)) \\ &+ \sum_{j=i+1}^{n-1} p_{O^*_{j+1}} d(\pi(j+1), O^*_j),\end{aligned}$$

and

$$\begin{aligned}\mathcal{S}_+(O^*) &= \sum_{j=1}^{i-2} p_{O^*_{j+1}} d(\pi(j+1), O^*_j) + p_{\pi(i)} p_{O^*_{i-1}} d(\pi(i), O^*_{i-1}) \\ &+ p_{\pi(i)} p_{\pi(i+1)} p_{O^*_{i-1}} d(\pi(i+1), O^*_{i-1} \cup \pi(i)) \\ &+ \sum_{j=i+1}^{n-1} p_{O^*_{j+1}} d(\pi(j+1), O^*_j).\end{aligned}$$

Since $O^* = \arg\max \mathcal{S}_+(O)$ is an optimal sequence, it must hold that $\mathcal{S}_+(O^*) - \mathcal{S}_+(O') \geq 0$, that is,

$$\begin{aligned}\mathcal{S}_+(O^*) - \mathcal{S}_+(O') &= p_{\pi(i)} p_{O^*_{i-1}} d(\pi(i), O^*_{i-1}) \\ &- p_{\pi(i+1)} p_{O^*_{i-1}} d(\pi(i+1), O^*_{i-1}) \\ &+ p_{\pi(i)} p_{\pi(i+1)} p_{O^*_{i-1}} d(\pi(i+1), O^*_{i-1}) \\ &- p_{\pi(i)} p_{\pi(i+1)} p_{O^*_{i-1}} d(\pi(i), O^*_{i-1}) \geq 0.\end{aligned}$$

This is equivalent to

$$\begin{aligned}&p_{\pi(i)} p_{O^*_{i-1}} (1 - p_{\pi(i+1)}) d(\pi(i), O^*_{i-1}) \\ &\geq p_{\pi(i+1)} p_{O^*_{i-1}} (1 - p_{\pi(i)}) d(\pi(i+1), O^*_{i-1}).\end{aligned} \tag{11.5}$$

Since $p_{O^*_{i-1}} > 0$, we can cancel that term from both sides of Equation (11.5). We then divide both sides by $(1 - p_{\pi(i)}) \times (1 - p_{\pi(i+1)})$ and obtain

$$\frac{p_{\pi(i)}}{1 - p_{\pi(i)}} d(\pi(i), O^*_{i-1}) \geq \frac{p_{\pi(i+1)}}{1 - p_{\pi(i+1)}} d(\pi(i+1), O^*_{i-1}). \tag{11.6}$$

By taking the telescope sum of Equation (11.6) over i from $i = 2$ to $i = j$, we get

$$\sum_{i=2}^{j} \frac{p_{\pi(i)}}{1 - p_{\pi(i)}} d(\pi(i), \pi(i-1)) \geq \frac{p_{\pi(j+1)}}{1 - p_{\pi(j+1)}} d(\pi(j+1), O^*_{j-1}). \tag{11.7}$$

By adding $\frac{p_{\pi(j+1)}}{1-p_{\pi(j+1)}} d(\pi(j+1), \pi(j))$ on both sides of Equation (11.7), we get

$$\sum_{i=2}^{j+1} \frac{p_{\pi(j)}}{1 - p_{\pi(i)}} d(\pi(i), \pi(i-1)) \geq \frac{p_{\pi(j+1)}}{1 - p_{\pi(j+1)}} d(\pi(j+1), O^*_j). \tag{11.8}$$

After moving $\frac{p_{\pi(j+1)}}{1-p_{\pi(j+1)}}$ to the left hand side of Equation (11.8) and exchanging the indexes of i and j, we get

$$d(\pi(i+1), O_i^*) \leq \frac{1-p_{\pi(i+1)}}{p_{\pi(i+1)}} \sum_{j=1}^{i} \frac{p_{\pi(j+1)}}{1-p_{\pi(j+1)}} d(\pi(j), \pi(j+1)),$$

which completes the proof. □

Corollary 4. *For an optimal sequence $O^* = \arg\max \mathcal{S}_+(O)$ to the MaxSSD problem, and assuming that $p_i \in [a,b]$, for all $i \in U$, it holds that $\mathcal{S}_+(O^*) \leq \frac{b(1-a)}{a(1-b)} \mathcal{H}(O^*)$.*

Proof. Let $O^* = (\pi(i))_{i=1}^n$ denote an optimal sequence. To proof this corollary it suffices to show that $d(\pi(i+1), O_i^*) \leq \frac{b(1-a)}{a(1-b)} d_L(O_{i+1}^*)$, for all $i \in [n-1]$.

Since we assume that $p_i \in [a,b]$, for all $i \in [n]$, we get

$$\frac{1-p_{\pi(i+1)}}{p_{\pi(i+1)}} \sum_{j=1}^{i} \frac{p_{\pi(j+1)}}{1-p_{\pi(j+1)}} d(\pi(j), \pi(j+1)) \leq \frac{b(1-a)}{a(1-b)} d_L(O_{i+1}^*). \quad (11.9)$$

Substituting Equation (11.9) into Lemma 5 leads to

$$d(\pi(i+1), O_i^*) \leq \frac{1-p_{\pi(i+1)}}{p_{\pi(i+1)}} \sum_{j=1}^{i} \frac{p_{\pi(j+1)}}{1-p_{\pi(j+1)}} d(\pi(j), \pi(j+1)) \leq \frac{b(1-a)}{a(1-b)} d_L(O_{i+1}^*),$$

which completes the proof. □

12 Additional preliminaries

In this section, we list a few useful lemmas that we are going to use repeatedly in the following sections. Since they are simple algebraic statements, we state them directly, without proof.

Lemma 6. *Let s_n be the sum of the the first n terms of a geometric series, namely $s_n = \sum_{i=0}^{n-1} ar^i$, where a is a constant, and r is any number such $0 < r < 1$. Then, $s_n = a\frac{1-r^n}{1-r}$.*

Lemma 7. *When r is a constant, $\lim_{n\to\infty} s_n = \frac{a}{1-r}$. In other words, when $n \to \infty$, it is $s_n = \frac{a}{1-r} - \Theta(r^n)$.*

Lemma 8 (Theorem 4.3, Chebyshev's inequality [12])**.** *Let $a_1 \leq \ldots \leq a_n$ and $b_1 \leq \ldots b_n$ be real numbers. Then, it holds*

$$\left(\sum_{i=1}^{n} a_i\right)\left(\sum_{i=1}^{n} b_i\right) \leq n \sum_{i=1}^{n} a_i b_i,$$

and equality holds when $a_1 = \ldots = a_n$ or $b_1 = \ldots = b_n$.

Lemma 9. $\left(1-\frac{1}{n}\right)^n = \frac{1}{e} - \Theta\left(\frac{1}{n}\right)$.

13 Omitted proofs from Section 6

Theorem 3. *Assume that $p_i = p < 1$, for all $i \in U$. For any integer τ, where $2 \le \tau \le n$, the BτI algorithm obtains a $(1 - p^{\tau-1} - p^{n-\tau} + p^n)$-approximation for MaxOHP.*

Proof. We start by introducing some additional notation. Let π be the ordering of U obtained from Algorithm 1, and O be the corresponding sequence. By definition,

$$\mathcal{H}(O) = \sum\nolimits_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1)).$$

Let π^o be the optimal ordering for MaxOHP, and O^o be the corresponding optimal sequence. Hence,

$$\mathcal{H}(O^o) = \sum\nolimits_{i=1}^{n-1} W_{O^o_i} d(\pi^o(i), \pi^o(i+1)).$$

Since $p_i = p$, for all $i \in U$, we notice that $p_{O_j} = p_{O^o_j} = p^j$. Hence, $W_{O_i} = W_{O^o_i} = \sum_{j=i+1}^{n} p^j$, after expansion, we get

$$\begin{aligned}\sum_{j=i+1}^{n} p^j &\overset{(a)}{=} p^{i+1}\frac{1-p^{n-i}}{1-p} \\ &= \frac{p^{i+1}}{1-p} - \frac{p^{n+1}}{1-p},\end{aligned}$$

where equality (a) holds by Lemma 6.

Recall that the BτI algorithm finds the optimal τ-item sequence that maximizes the following equation:

$$O_\tau = \operatorname*{arg\,max}_{O_\tau \sqsubseteq \pi(U)} \sum_{i=1}^{\tau-1} \frac{p^{i+1}}{1-p} d(\pi(i), \pi(i+1)). \tag{13.1}$$

For the convenience in our calculations, we let

$$\ell_\pi = \sum_{i=1}^{\tau-1} \frac{p^{i+1}}{1-p} d(\pi(i), \pi(i+1)).$$

Let us derive a lower bound on $\mathcal{H}(O)$ as follows:

$$\begin{aligned}\mathcal{H}(O) &= \sum_{i=1}^{n-1} \frac{p^{i+1}}{1-p} d(\pi(i), \pi(i+1)) - \sum_{i=1}^{n-1} \frac{p^{n+1}}{1-p} d(\pi(i), \pi(i+1)) \\ &\ge \sum_{i=1}^{\tau-1} \frac{p^{i+1}}{1-p} d(\pi(i), \pi(i+1)) - \sum_{i=1}^{\tau-1} \frac{p^{n+1}}{1-p} d(\pi(i), \pi(i+1)) \\ &= \ell_\pi - \frac{p^{n+1}}{1-p} \sum_{i=1}^{\tau-1} d(\pi(i), \pi(i+1)).\end{aligned} \tag{13.2}$$

Next, we derive an upper bound on $\mathcal{H}(O^o)$. Let $T = \lceil \frac{n}{\tau-1} \rceil$,

$$\begin{aligned}\mathcal{H}(O^o) &= \sum_{i=1}^{n-1} \frac{p^{i+1}}{1-p} d(\pi^o(i), \pi^o(i+1)) - \sum_{i=1}^{n-1} \frac{p^{n+1}}{1-p} d(\pi^o(i), \pi^o(i+1)) \\ &\le \sum_{i=1}^{n-1} \frac{p^{i+1}}{1-p} d(\pi^o(i), \pi^o(i+1)) \\ &\le \sum_{t=1}^{T} \sum_{i=(t-1)(\tau-1)+1}^{t\cdot(\tau-1)} \frac{p^{i+1}}{1-p} d(\pi^o(i), \pi^o(i+1)) \\ &\overset{(a)}{\le} \ell_\pi (1 + p^{\tau-1} + \cdots + p^{(T-1)(\tau-1)}) \\ &\le \frac{\ell_\pi}{1-p^{\tau-1}}.\end{aligned} \tag{13.3}$$

Notice that (a) holds due to the optimality of ℓ_π. Notice that $T\cdot(\tau-1)$ might be greater than n, we can hypothetically set $d(\pi^o(i), \pi^o(i+1)) = 0$ for $i \ge n$, for our analysis.

Combining Equations (13.2) and (13.3), we get

$$\begin{aligned}\frac{\mathcal{H}(O)}{\mathcal{H}(O^o)} &\ge (1-p^{\tau-1}) \frac{\ell_\pi - \frac{p^{n+1}}{1-p}\sum_{i=1}^{\tau-1} d(\pi(i), \pi(i+1))}{\ell_\pi} \\ &= (1-p^{\tau-1}) \left(1 - \frac{\frac{p^{n+1}}{1-p}\sum_{i=1}^{\tau-1} d(\pi(i), \pi(i+1))}{\sum_{i=1}^{\tau-1} \frac{p^{i+1}}{1-p} d(\pi(i), \pi(i+1))}\right) \\ &\ge (1-p^{\tau-1}) \left(1 - \frac{\frac{p^{n+1}}{1-p}\sum_{i=1}^{\tau-1} d(\pi(i), \pi(i+1))}{\sum_{i=1}^{\tau-1} \frac{p^{\tau}}{1-p} d(\pi(i), \pi(i+1))}\right) \\ &= (1-p^{\tau-1})(1-p^{n-\tau+1}) \\ &= 1 - p^{\tau-1} - p^{n-\tau+1} + p^n \\ &\ge 1 - p^{\tau-1} - p^{n-\tau} + p^n.\end{aligned}$$

We have demonstrated that the BτI algorithm provides an asymptotic $(1 - p^{\tau-1} - p^{n-\tau} + p^n)$-approximation. □

13.1 Proof of Theorem 4

To prove Theorem 4, we first analyze the property of the greedy matching M obtained from Algorithm 2.

Lemma 10. *Given a graph $G = (V, E)$ with $|V| = n$, let $w(S)$ denote the sum of edge weights in an edge subset $S \subseteq E$. Let M be the greedy matching of G, M_k denote the first k edges of M, and M^*_k be the maximum weighted size k matching of G. Then it holds that $w(M_k) \ge \frac{1}{2} w(M^*_k)$ for all $k \le \lfloor \frac{n}{2} \rfloor$.*

Proof. Assume there exists an edge $e \in M^*_k \setminus M_k$, then there must exist an edge $e' \in M_k$ such that e' share one end node with e and $w(e) < w(e')$, because otherwise the greedy matching algorithm will choose e instead of e'. Since each such e' can conflict at most two edges from M^*_k, we have that

$$2 \sum_{e' \in M_k \setminus M^*_k} w(e') > \sum_{e \in M^*_k \setminus M_k} w(e),$$

and the lemma thus follows. □

Theorem 4. *Assume that $p_i = p$, for any $i \in U$. Let p be a function of n such that $\lim_{n\to\infty} p = 1$ and $\lim_{n\to\infty} p^n = 0$. Moreover, assume that $p = 1 - \frac{1}{t_n}$, where t_n is a increasing function of n and $t_n = o(n)$. The algorithm GM yields a $\left(\frac{3(e-1)}{16e^2} - \Theta\left(\frac{1}{t_n}\right)\right)$-approximation for MaxOHP.*

Proof. We let π be the ordering of U obtained from GM algorithm, and O be the corresponding sequence. Recall that

$$\mathcal{H}(O) = \sum_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1)). \tag{13.4}$$

Since p^i decreases as i increases, we have $p^i \geq p^{t_n}$, for all $i \leq t_n$. Hence, we obtain a simple lower bound on W_{O_i} for any $i \leq t_n$.

$$W_{O_i} = \sum_{j=i+1}^{n} p^j \geq \sum_{j=i+1}^{t_n} p^j \geq (t_n - i)p^{t_n}. \tag{13.5}$$

Recall that, the π we choose satisfies the following two properties, which we are soon going to use to obtain a lower bound of $\mathcal{H}(O)$.

(P1) $d((\pi(2i-1), \pi(2i))) \geq d((\pi(2i+1), \pi(2i+2)))$ for $i < \lfloor \frac{n}{2} \rfloor$, and

(P2) $d((\pi(2i), \pi(2i+1))) \geq \frac{1}{2} d((\pi(2i-1), \pi(2i)))$ for $i \leq \lfloor \frac{n}{2} \rfloor$.

Let us simplify $\mathcal{H}(O)$ by applying the above observation. In the following analysis, we define $\mathcal{T}_n := \lfloor (t_n - 1)/2 \rfloor$.

$$\begin{aligned}
\mathcal{H}(O) &\overset{(a)}{\geq} \sum_{i=1}^{\mathcal{T}_n} \left(W_{O_{2i-1}} d(\pi(2i-1), \pi(2i)) + W_{O_{2i}} d(\pi(2i), \pi(2i+1)) \right) \\
&\overset{(b)}{\geq} \sum_{i=1}^{\mathcal{T}_n} \left(W_{O_{2i-1}} + \frac{1}{2} W_{O_{2i}} \right) d(\pi(2i-1), \pi(2i)) \\
&\overset{(c)}{\geq} \frac{1}{\mathcal{T}_n} \sum_{i=1}^{\mathcal{T}_n} \left(W_{O_{2i-1}} + \frac{1}{2} W_{O_{2i}} \right) \sum_{i=1}^{\mathcal{T}_n} d(\pi(2i-1), \pi(2i)) \\
&\overset{(d)}{\geq} \frac{p^{t_n}}{\mathcal{T}_n} \sum_{i=1}^{\mathcal{T}_n} \left((t_n - 2i + 1) + \frac{1}{2}(t_n - 2i) \right) \sum_{i=1}^{\mathcal{T}_n} d(\pi(2i-1), \pi(2i)) \\
&= \frac{p^{t_n}}{\mathcal{T}_n} \sum_{i=1}^{\mathcal{T}_n} \left(-3i + 1 + \frac{3}{2} t_n \right) \sum_{i=1}^{\mathcal{T}_n} d(\pi(2i-1), \pi(2i)) \\
&= \left(p^{t_n}\right) \left(\frac{-3 - 3\mathcal{T}_n}{2} + 1 + \frac{3}{2} t_n \right) \sum_{i=1}^{\mathcal{T}_n} d(\pi(2i-1), \pi(2i)).
\end{aligned} \tag{13.6}$$

Inequality (a) holds as we decompose $\mathcal{H}(O)$ into two parts, the series of $W_{O_{2i-1}} d(\pi(2i-1), \pi(2i))$ are related with the top-$\mathcal{T}_n$ matching, and the series of $W_{O_{2i}} d(\pi(2i), \pi(2i+1))$ are related with the edges that connect this matching.

Inequality (b) holds according to the property **(P2)**.

Inequality (c) holds as we apply Lemma 8 by property **(P1)**.

Inequality (d) holds as we reformulate $W_{O_{2i-1}}$ and $W_{O_{2i}}$ according to Equation (13.5).

Let ℓ_{t_n} denote the largest weights of a path that consists of t_n nodes chosen from U, i.e.,

$$\ell_{t_n} = \max_{\pi} \sum_{i=1}^{t_n - 1} d(\pi(i), \pi(i+1)). \tag{13.7}$$

Notice that as $\{\pi(2i-1), \pi(2i)\}_{i=1}^{\mathcal{T}_n}$ is a greedy matching of the first $\mathcal{T}_n$ edges, by Lemma 10, it contains at least $\frac{1}{2}$ weights of the maximum weighted size $\mathcal{T}_n$ matching. Hence, it consists of at least $\frac{1}{4}$ of the largest weights of the path that consists of $2\mathcal{T}_n$ edges. We can get

$$\sum_{i=1}^{\mathcal{T}_n} d(\pi(2i-1), \pi(2i)) \geq \frac{1}{4} \frac{t_n - 2}{t_n - 1} \ell_{t_n},$$

where the factor $\frac{t_n - 2}{t_n - 1}$ is added as $2\mathcal{T}_n$ can either be $t_n - 1$ or $t_n - 2$.

We substitute the above result into Equation (13.6), and we get

$$\begin{aligned}
\mathcal{H}(O) &\geq \left(p^{t_n}\right) \left(\frac{-3 - 3\mathcal{T}_n}{2} + 1 + \frac{3}{2} t_n \right) \frac{1}{4} \frac{t_n - 2}{t_n - 1} \ell_{t_n} \\
&\overset{(a)}{>} \left(p^{t_n}\right) \left(\frac{3}{4} t_n - \frac{1}{2} \right) \frac{1}{4} \frac{t_n - 2}{t_n - 1} \ell_{t_n} \\
&= \left(p^{t_n}\right) \left(\frac{3}{16} t_n - \frac{1}{8} \right) \frac{t_n - 2}{t_n - 1} \ell_{t_n},
\end{aligned} \tag{13.8}$$

where inequality (a) holds by substituting an upper bound $\frac{t_n}{2}$ of $\mathcal{T}_n$ into $\mathcal{T}_n$,

We let π^o be the optimal ordering for the MaxOHP, and O^o be the corresponding optimal sequence. Hence,

$$\mathcal{H}(O^o) = \sum_{i=1}^{n-1} W_{O_i^o} d(\pi^o(i), \pi^o(i+1)).$$

We can obtain a simple upper bound on $W_{O_i^o}$ by

$$\begin{aligned}
W_{O_i^o} &= \sum_{j=i+1}^{n} p^j \\
&= \frac{p^{i+1}}{1-p} - \frac{p^{n+1}}{1-p} \\
&\leq \frac{p^{i+1}}{1-p} \\
&\overset{(a)}{=} p^{i+1} t_n \\
&\leq p^{\lfloor \frac{i+1}{t_n} \rfloor \cdot t_n} t_n,
\end{aligned} \tag{13.9}$$

where equality (a) holds as we can use $p = 1 - \frac{1}{t_n}$.

By letting $T = \left\lceil \frac{n}{t_n - 1} \right\rceil$ we can derive an upper bound of $\mathcal{H}(O^o)$ as follows

$$\begin{aligned}
\mathcal{H}(O^o) &\leq \sum_{t=0}^{T-1} \sum_{i=t\cdot(t_n-1)+1}^{(t+1)\cdot(t_n-1)} W_{A_i^o} d(\pi^o(i), \pi^o(i+1)) \\
&\overset{(a)}{\leq} \sum_{t=0}^{T-1} \sum_{i=t\cdot(t_n-1)+1}^{(t+1)\cdot(t_n-1)} p^{t\cdot t_n} t_n d(\pi^o(i), \pi^o(i+1)) \\
&\overset{(b)}{\leq} t_n \cdot \ell_{t_n} \sum_{t=0}^{T-1} p^{t\cdot t_n} \\
&\leq \frac{t_n \cdot \ell_{t_n}}{1 - p^{t_n}}.
\end{aligned} \tag{13.10}$$

where inequality (a) holds because of Equation (13.9) and inequality (b) holds because of Equation (13.7). We set $d(\pi^o(i), \pi^o(i+1)) = 0$ for $i \geq n$ for the first inequality to hold.

Combining Equations (13.8) and (13.10) leads to

$$\frac{\mathcal{H}(O)}{\mathcal{H}(O^o)} \geq \frac{(p^{t_n}) \cdot \left(\frac{3}{16}t_n - \frac{1}{8}\right)\frac{t_n-2}{t_n-1}\ell_{t_n}}{\frac{t_n \cdot \ell_{t_n}}{1-p^{t_n}}}$$
$$= p^{t_n}(1-p^{t_n}) \cdot \left(\frac{3}{16} - \frac{1}{8t_n}\right)\frac{t_n-2}{t_n-1}$$
$$= p^{t_n}(1-p^{t_n}) \cdot \left(\frac{3}{16} - \Theta\left(\frac{1}{t_n}\right)\right)\left(1-\Theta\left(\frac{1}{t_n}\right)\right)$$
$$\overset{(a)}{=} \left(\frac{e-1}{e} + \Theta\left(\frac{1}{t_n}\right)\right)\left(\frac{1}{e} - \Theta\left(\frac{1}{t_n}\right)\right)\left(\frac{3}{16} - \Theta\left(\frac{1}{t_n}\right)\right)\left(1-\Theta\left(\frac{1}{t_n}\right)\right)$$
$$= \frac{3(e-1)}{16e^2} - \Theta\left(\frac{1}{t_n}\right),$$

where (a) holds as we apply Lemma 9, and use

$$p^{t_n} = (1-\frac{1}{t_n})^{t_n} = \frac{1}{e} - \Theta\left(\frac{1}{t_n}\right).$$

Thus we have proven that GM is a $\left(\frac{3(e-1)}{16e^2} - \Theta\left(\frac{1}{t_n}\right)\right)$-approximation algorithm.

□

13.2 The case of an arbitrary ranking

Let us consider a very special case where the user examines all the items. In particular, we assume that $\lim_{n\to\infty} p = 1$ and $\lim_{n\to\infty} p^n = c$, where c is a non-zero constant. For this special case, we notice that the order of items no longer matters — even an arbitrary order will provide a constant-factor approximation to our problem. Intuitively, this is because for such a large value of p the user will examine all the items of U. Thus, we can apply Lemma 1, and obtain a constant-factor approximation algorithm for problem MaxSSD directly, without considering MaxOHP.

Theorem 6. *Assume that $p_i = p$, for all $i \in U$. Moreover, assume that $\lim_{n\to\infty} p = 1$ and $p^n = c - \frac{1}{\Theta(n)}$, where c is a constant. Then any ordering of the items of U yields a $\left(c - \frac{1}{\Theta(n)}\right)$-approximation for the MaxSSD problem.*

Proof. We let π be any ordering of U, and O be the corresponding sequence. We let π^* be the optimal order for the problem MaxSSD, and let O^* be the corresponding sequence. We have

$$\mathcal{S}_+(O) = \sum_{i=1}^{n-1} p_{O_{i+1}} d(\pi(i+1), O_i)$$
$$\geq \sum_{i=1}^{n-1}\left(c - \Theta\left(\frac{1}{n}\right)\right) d(\pi(i+1), O_i)$$
$$= \left(c - \Theta\left(\frac{1}{n}\right)\right)\sum_{i=1}^{n-1} d(\pi(i+1), O_i)$$
$$\overset{(a)}{\geq} \left(c - \Theta\left(\frac{1}{n}\right)\right)\mathcal{S}_+(O^*),$$

where inequality (a) holds because $\mathcal{S}_+(O^*)$ is not larger than the sum of all pairs of diversities.

Hence,

$$\frac{\mathcal{S}_+(O)}{\mathcal{S}_+(O^*)} \geq c - \frac{1}{\Theta(n)}.$$

□

14 Omitted proofs from Section 7

Theorem 5. *Let $p_i \in [a,b]$ for all $i \in U$, where $0 < a \leq b < 1$. Let τ be an integer such that $2 \leq \tau \leq n$. The BτI algorithm provides a $\frac{a^2(1-b)(1-b^{\tau-1})}{a^2+(\tau-1)b^{\tau+1}}$-approximation guarantee for MaxOHP.*

Proof. Let π be the ordering of U obtained from our algorithm. Let $O_\pi = (\pi(i))_{i=1}^n$ denote the ordered sequence according to π and write $O = O_\pi$. Recall that

$$\mathcal{H}(O) = \sum_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1)).$$

Let π^o be the optimal ordering for the MaxOHP. Let $O_{\pi^o} = (\pi^o(i))_{i=1}^n$ and write $O^o = O_{\pi^o}$. For the optimal ordering O^o, let $\mathcal{H}(O^o) = \sum_{i=1}^{n-1} W_{O_i^o} d(\pi^o(i), \pi^o(i+1))$.

Recall that

$$W_{O_i} := \sum_{j=i+1}^{n} p_{O_j}.$$

Hence,

$$\mathcal{H}(O) = \sum_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i+1))$$
$$\geq \sum_{i=1}^{\tau-1} W_{O_i} d(\pi(i), \pi(i+1))$$
$$\geq \sum_{i=1}^{\tau-1}\sum_{j=i+1}^{\tau} p_{O_j} d(\pi(i), \pi(i+1)).$$

We define

$$\ell_\tau = \sum_{i=1}^{\tau-1}\sum_{j=i+1}^{\tau} p_{O_j} d(\pi(i), \pi(i+1)).$$

It is not hard to see that ℓ_τ is the function value maximized in Equation (7.1). By setting $T = \left\lceil \frac{n-1}{\tau-1} \right\rceil$, we can rewrite $\mathcal{H}(O^o)$ as follows

$$\mathcal{H}(O^o) = \sum_{i=1}^{n-1} W_{O_i^o} d(\pi^o(i), \pi^o(i+1))$$
$$= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} p_{O_j^o} d(\pi^o(i), \pi^o(i+1))$$
$$= \sum_{t=1}^{T}\sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)}\sum_{j=i+1}^{n} p_{O_j^o} d(\pi^o(i), \pi^o(i+1))$$
$$= \sum_{t=1}^{T}\sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)}\left(\sum_{j=i+1}^{t\cdot\tau} p_{O_j^o} + \sum_{j=t\cdot\tau+1}^{n} p_{O_j^o}\right) d(\pi^o(i), \pi^o(i+1))$$

Notice that for each fixed t, $\mathcal{H}(O^o)$ is the sum of two parts:

$$\sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)}\sum_{j=i+1}^{t\cdot\tau} p_{O_j^o} d(\pi^o(i), \pi^o(i+1)) \qquad (14.1)$$

$$\sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)} \sum_{j=t\cdot\tau+1}^{n} p_{O_j^o} d(\pi^o(i), \pi^o(i+1)) \tag{14.2}$$

Let us first discuss the part (14.1)

$$\begin{aligned}\sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)} \sum_{j=i+1}^{t\cdot\tau} p_{O_j^o} d(\pi^o(i), \pi^o(i+1)) \\ \overset{(a)}{\leq} \ell_\tau p_{O^o_{(t-1)(\tau-1)}} \\ \leq \ell_\tau b^{(t-1)(\tau-1)}.\end{aligned} \tag{14.3}$$

Similarly, for part (14.2)

$$\begin{aligned}&\sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)} \sum_{j=t\cdot\tau+1}^{n} p_{O_j^o} d(\pi^o(i), \pi^o(i+1)) \\ &\overset{(b)}{\leq} \sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)} \sum_{j=t\cdot\tau+1}^{n} b^j d(\pi^o(i), \pi^o(i+1)) \\ &\leq \sum_{i=(t-1)(\tau-1)+1}^{t(\tau-1)} \frac{b^{t\cdot\tau+1}}{1-b} d(\pi^o(i), \pi^o(i+1)) \\ &\leq (\tau-1) d_{\max} \frac{b^{t\cdot\tau+1}}{1-b},\end{aligned} \tag{14.4}$$

where $d_{\max} \coloneqq \max_{u,v\in U} d(u,v)$. Inequality (a) holds by the definition of ℓ_τ and our algorithm, and inequality (b) holds because $p_{O_j^o} \leq b^j$.

We combine Equations (14.3) and (14.4) and we continue to bound $\mathcal{H}(O^o)$:

$$\begin{aligned}\mathcal{H}(O^o) &\leq \sum_{t=1}^{T} \left(\ell_\tau b^{(t-1)(\tau-1)} + (\tau-1) d_{\max} \frac{b^{t\cdot\tau+1}}{1-b} \right) \\ &\leq \frac{\ell_\tau}{1-b^{\tau-1}} + (\tau-1) d_{\max} \frac{b^{\tau+1}}{(1-b)(1-b^\tau)},\end{aligned} \tag{14.5}$$

Next, we derive an upper bound of the second term of Equation (14.5) in terms of ℓ_τ. Let $\hat{\pi}$ be any order, which places the edge with edge weight $d_{\max}$ to the first two positions. Let $\hat{O}$ be the corresponding sequence. We have:

$$\ell_\tau \overset{(a)}{\geq} \ell_2 \overset{(b)}{\geq} p_{\hat{O}_2} d(\hat{\pi}(1), \hat{\pi}(2)) \geq a^2 d_{\max}. \tag{14.6}$$

Notice that inequality (a) holds as ℓ_τ should consist of at least one edge, and inequality (b) holds because of the optimality of ℓ_τ. Thus,

$$\begin{aligned}\frac{\mathcal{H}(O)}{\mathcal{H}(O^o)} &\geq \frac{1}{\frac{1}{1-b^{\tau-1}} + \frac{\tau-1}{a^2} \cdot \frac{b^{\tau+1}}{(1-b)(1-b^\tau)}} \\ &= \frac{a^2(1-b^{\tau-1})(1-b)(1-b^\tau)}{a^2(1-b)(1-b^\tau) + (\tau-1)b^{\tau+1}(1-b^{\tau-1})} \\ &> \frac{a^2(1-b^{\tau-1})(1-b)(1-b^\tau)}{a^2(1-b)(1-b^\tau) + (\tau-1)b^{\tau+1}(1-b^\tau)} \\ &= \frac{a^2(1-b^{\tau-1})(1-b)}{a^2(1-b) + (\tau-1)b^{\tau+1}} \\ &> \frac{a^2(1-b)(1-b^{\tau-1})}{a^2 + (\tau-1)b^{\tau+1}}\end{aligned}$$

This completes the proof. □

15 Experiments and results

In this section, we present the omitted dataset information, baseline description, the experimental setting, and more experimental results.

15.1 Datasets

Coat:[2] Coat ratings in the range [1, 5], accompanied by categorical features, such as gender, jacket type, and color.
KuaiRec [14]: Recommendation logs from a video-sharing mobile app, containing each video's duration, categories, and user viewing times. The *watch ratio*, calculated as the user's viewing time divided by the video's duration, and normalized to range [1, 5], serves as personalized rating.
Netflix:[3,4] Movie ratings in range [1, 5] with genre information. We sample 5 K movies and exclude users with less than 20 interactions.
Movielens:[5] Movie ratings in range [1, 5] with movie genres.
Yahoo:[6] Song ratings in range [1, 5] with song genres. We sample 3 K songs and exclude users with less than 20 interactions.
LETOR:[7] Web-search dataset with relevance scores (in $\{0, 1, 2\}$) on queries-document pairs and document feature vectors.
LTRC:[8] Web-search dataset with relevance scores (in $\{0, 1, 2, 3, 4\}$) on queries-document pairs and document feature vectors.

15.2 Omitted Baseline

EXPLORE [11]. The EXPLORE algorithm iteratively generates a size-k recommendation list through a greedy selection process, followed by a simulation to collect the items accepted by the user. In each iteration, EXPLORE selects the top k items from $U \setminus R$ that maximizes the following expression:

$$[p_i^{-\alpha} + d(i,R)^{-\alpha} - 1]^{-1/\alpha}.$$

After presenting the recommendation list to the user, they either accept one of the items and proceed to the next iteration, or they quit with certain probability. If the user does not quit, EXPLORE adds the accepted item to R and continues the process in the next iteration.

EXPLORE studies a setting distinct from ours, as outlined previously. In EXPLORE, multiple recommendation lists are generated, and users can select one item from each list. In contrast, our setting involves a single ordering, where users examine items sequentially in the predefined order. To adapt EXPLORE to our setting, we propose two methods for generating the ordering.

- The first method treats the set R as the ordered sequence, preserving the order in which items are included, and appends the remaining items from $U \setminus R$ randomly to the end of the sequence.

[2] https://www.cs.cornell.edu/~schnabts/mnar/
[3] https://www.kaggle.com/datasets/rishitjavia/netflix-movie-rating-dataset
[4] https://github.com/tommasocarraro/netflix-prize-with-genres
[5] https://grouplens.org/datasets/movielens/1m/
[6] https://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67
[7] https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/letor-4-0/
[8] https://webscope.sandbox.yahoo.com/catalog.php?datatype=c

Table 4: Sequential sum diversity $\mathcal{S}_+(O)$ with item continuation probability mapped to $[0.1, 0.3]$.The results marked with * are obtained using the BτI-H heuristic

	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
Coat	0.039 ± 0.020	0.105 ± 0.041	0.102 ± 0.038	0.095 ± 0.039	0.105 ± 0.041	0.105 ± 0.041	**0.109 ± 0.042**	0.106 ± 0.038	0.104* ± 0.037*
KuaiRec	0.022 ± 0.006	0.050 ± 0.000	0.033 ± 0.008	0.042 ± 0.006	0.051 ± 0.010	0.044 ± 0.006	**0.058 ± 0.010**	**0.058 ± 0.009***	**0.058 ± 0.009***
Netflix	0.063 ± 0.026	0.152 ± 0.026	0.145 ± 0.026	0.161 ± 0.024	0.162 ± 0.024	0.162 ± 0.024	**0.163 ± 0.024**	0.160 ± 0.025	0.159* ± 0.025*
Movielens	0.059 ± 0.028	0.173 ± 0.013	0.169 ± 0.018	**0.183 ± 0.003**	**0.183 ± 0.003**	**0.183 ± 0.003**	**0.183 ± 0.003**	**0.183 ± 0.003***	**0.183 ± 0.003***
Yahoo	0.016 ± 0.027	0.133 ± 0.044	**0.148 ± 0.044**	**0.148 ± 0.044**	**0.148 ± 0.044**	**0.148 ± 0.044**	**0.148 ± 0.044**	0.140 ± 0.045*	0.139 ± 0.045*
LETOR	0.008 ± 0.008	0.026 ± 0.021	**0.043 ± 0.028**	0.039 ± 0.026	0.040 ± 0.026	0.040 ± 0.026	**0.043 ± 0.028**	0.042 ± 0.027	0.042 ± 0.027
LTRC	0.019 ± 0.011	0.041 ± 0.018	0.054 ± 0.020	0.051 ± 0.020	0.051 ± 0.019	0.051 ± 0.019	**0.055 ± 0.020**	0.054 ± 0.020	0.054 ± 0.020

Table 5: Sequential sum diversity $\mathcal{S}_+(O)$ with item continuation probability mapped to $[0.7, 0.9]$.The results marked with * are obtained using the BτI-H heuristic

	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
Coat	9.703 ± 3.704	23.044 ± 12.390	22.251 ± 11.328	18.586 ± 8.481	24.151 ± 13.421	**24.213 ± 13.748**	23.030 ± 12.385	22.839 ± 12.428*	22.826 ± 12.415*
KuaiRec	5.737 ± 1.094	10.668 ± 1.874	11.599 ± 3.289	9.119 ± 1.090	**14.654 ± 3.874**	12.565 ± 4.258	10.920 ± 1.728	10.906 ± 1.714*	10.968 ± 1.726*
Netflix	16.901 ± 5.588	47.316 ± 14.231	39.696 ± 9.856	35.729 ± 10.010	48.595 ± 14.424	**50.530 ± 15.407**	47.224 ± 14.648	47.053 ± 14.294*	47.172 ± 14.266*
Movielens	14.677 ± 4.644	65.410 ± 6.715	48.412 ± 4.246	63.242 ± 8.458	69.341 ± 7.604	**71.362 ± 7.006**	71.303 ± 7.748	71.305 ± 7.738*	71.317 ± 7.703*
Yahoo	4.014 ± 3.259	40.505 ± 19.773	38.251 ± 18.866	38.520 ± 19.221	39.134 ± 19.191	38.377 ± 19.021	41.677 ± 21.090	41.924 ± 20.996*	**41.969 ± 21.010***
LETOR	2.784 ± 1.638	5.825 ± 3.450	5.317 ± 2.755	6.140 ± 2.825	**6.523 ± 3.803**	6.025 ± 3.637	6.034 ± 3.006	6.086 ± 3.033	6.165 ± 3.048
LTRC	4.757 ± 1.339	7.026 ± 2.161	6.332 ± 1.691	7.088 ± 1.958	**7.385 ± 2.352**	7.204 ± 2.164	6.972 ± 1.971	7.005 ± 1.990	7.058 ± 2.012

Table 6: Sequential sum diversity $\mathcal{S}_+(O)$ with item continuation probabilities mapped to $[0.1, 0.9]$.The results marked with * are obtained using the BτI-H heuristic

	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
Coat	0.555±0.722	7.665±10.628	8.409±10.708	9.660±13.374	**9.661±13.608**	9.559±13.534	9.075±12.593	9.119±12.67*7	9.100±12.686*
KuaiRec	0.255±0.191	1.140±0.554	4.808±3.282	6.388±4.981	**6.768±5.221**	6.739±5.140	3.726±2.575	3.745±2.570*	3.834±2.617*
Netflix	1.704±1.822	27.252±19.761	22.197±12.660	32.270±22.224	32.703±22.895	**33.157±23.552**	30.790±21.842	30.848±21.822*	30.915±21.813*
Movielens	1.157±1.283	61.736±9.103	37.570±5.342	59.667±13.080	68.657±9.789	66.995±13.290	69.476±11.786	69.478±11.780*	**69.484±11.759***
Yahoo	0.394±0.927	26.104±24.181	21.018±20.291	22.055±20.980	26.303±23.451	21.039±20.324	27.567±26.163	27.709±26.100*	**27.746±26.105***
LETOR	0.127 ± 0.137	0.436 ± 0.419	0.583 ± 0.478	0.595 ± 0.491	0.612 ± 0.521	0.592 ± 0.510	0.610 ± 0.515	0.608 ± 0.516	**0.615 ± 0.520**
LTRC	0.189 ± 0.216	0.702 ± 0.640	1.050 ± 0.865	1.096 ± 0.914	**1.136 ± 0.996**	**1.136 ± 1.000**	1.091 ± 0.899	1.099 ± 0.911	1.112 ± 0.928

- The second method concatenates the lists generated during each iteration to form the sequence. If the user exits, the remaining items are appended randomly to the sequence.

For parameter tuning of the EXPLORE algorithm, we set $\alpha = 0.5$ as recommended in [11], and vary the length of the recommendation list, k, over the set $[1, 5, 10, 20]$. Additionally, we vary the number of expected exploration steps over the set $[5, 10, 20]$. We evaluate the sequential diversity value and report the best performance for both adaptation methods along with the corresponding parameter settings.

15.3 More experimental results

Experimental environment All experiments are conducted on a dual-socket server, each socket housing an Intel Xeon Gold 6326 CPU at 2.90GHz, with 128 cores and 256 threads.

More experimental results We evaluate the performance of the proposed BτI algorithm and the BτI-H heuristic, by running these methods in a continuation probabilities set in small, large and full regime. We present our results in Table 4, Table 5 and Table 6. We also report the *Expnum*, *ExpDCG* and expected *ExpSerendipity* values when continuation probabilities are mapped to small, large and full regimes in Table 7, Table 8 and Table 9 respectively. We present the running time of our methods and the baselines on different datasets in Table 10. All results marked with * are obtained from the BτI-H heuristic

The EXPLORE algorithm, despite designed to maximize the sequential diversity, generally under-performs compared to BτI in terms of sequential diversity value. This may be due to the different settings, as EXPLORE is designed for scenarios where users interact with multiple lists, accepting only one item from each, whereas BτI allows users to sequentially accept items from a single list.

Table 7: ***Expnum*, *ExpDCG* and *ExpSerendipity* values with item continuation probabilities mapped to $[0.1, 0.3]$. The results with * are obtained using the BτI-H heuristic**

Metrics	Datasets	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
ExpDCG	Coat	0.040 ± 0.020	0.077±0.027	**0.082 ± 0.026**	0.078 ± 0.025	**0.082 ± 0.026**	**0.082 ± 0.025**	0.075 ± 0.026	0.076 ± 0.026*	0.076 ± 0.026*
	KuaiRec	0.040 ± 0.012	0.080±0.017	**0.101 ± 0.013**	0.085 ± 0.010	0.091 ± 0.014	0.085 ± 0.010	0.070 ± 0.023	0.090 ± 0.011*	0.091 ± 0.011*
	Netflix	0.054 ± 0.020	0.095±0.017	**0.105 ± 0.012**	**0.105 ± 0.012**	**0.105 ± 0.012**	**0.105 ± 0.012**	0.103 ± 0.012	0.104 ± 0.012*	0.104 ± 0.012*
	Movielens	0.054 ± 0.025	0.107±0.010	**0.112 ± 0.001**	**0.112 ± 0.001**	**0.112 ± 0.001**	**0.112 ± 0.001**	**0.112 ± 0.001**	**0.112 ± 0.001***	**0.112 ± 0.001***
	Yahoo	0.044 ± 0.026	0.090±0.027	**0.099 ± 0.022**	**0.099 ± 0.022**	**0.099 ± 0.022**	**0.099 ± 0.022**	0.095 ± 0.025	**0.099 ± 0.022***	**0.099 ± 0.022***
	LETOR	0.022 ± 0.022	0.030±0.028	0.050 ± 0.038	0.065 ± 0.036	0.065 ± 0.036	**0.066 ± 0.036**	0.050 ± 0.038	0.063 ± 0.035	0.063 ± 0.035
	LTRC	0.033 ± 0.019	0.042±0.024	0.056 ± 0.024	**0.070 ± 0.026**	**0.070 ± 0.026**	0.069 ± 0.025	0.056 ± 0.024	0.064 ± 0.025	0.064 ± 0.025
ExpSerendipity	Coat	0.013 ± 0.020	0.029±0.035	0.022 ± 0.029	0.017 ± 0.025	0.019 ± 0.026	0.019 ± 0.026	0.037 ± 0.038	**0.041 ± 0.041***	**0.041 ± 0.041***
	Netflix	0.005 ± 0.015	0.014±0.027	**0.019 ± 0.033**	**0.019 ± 0.032**	**0.019 ± 0.032**	**0.019 ± 0.032**	0.018 ± 0.031	0.017 ± 0.030*	0.017 ± 0.030*
	Movielens	0.006 ± 0.018	0.018±0.029	**0.019 ± 0.032**	0.013 ± 0.026	0.013 ± 0.026	0.013 ± 0.026	0.013 ± 0.026	0.013 ± 0.026*	0.013 ± 0.026*
	Yahoo	0.003 ± 0.012	0.020±0.021	0.021 ± 0.017	0.019 ± 0.015	0.019 ± 0.015	0.019 ± 0.015	**0.027 ± 0.023**	0.017 ± 0.014*	0.017 ± 0.014*

Table 8: ***Expnum*, *ExpDCG* and *ExpSerendipity* values with item continuation probabilities mapped to $[0.7, 0.9]$. The results with * are obtained using the BτI-H heuristic**

Metrics	Datasets	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
ExpDCG	Coat	1.564 ± 0.275	2.267±0.534	2.324 ± 0.532	2.101 ± 0.437	2.308 ± 0.540	**2.346 ± 0.555**	2.219 ± 0.508	2.233 ± 0.512*	2.232 ± 0.511*
	KuaiRec	1.565 ± 0.162	2.034±0.174	2.789 ± 0.308	1.772 ± 0.104	2.416 ± 0.443	**2.905 ± 0.350**	1.772 ± 0.145	1.887 ± 0.130*	1.891 ± 0.131*
	Netflix	1.889 ± 0.280	2.881±0.365	2.880 ± 0.308	2.681 ± 0.290	**3.003 ± 0.350**	2.993 ± 0.349	2.841 ± 0.338	2.855 ± 0.332*	2.865 ± 0.330*
	Movielens	1.782 ± 0.287	3.280±0.101	3.141 ± 0.069	3.213 ± 0.109	3.282 ± 0.097	**3.327 ± 0.075**	3.286 ± 0.097	3.287 ± 0.096*	3.288 ± 0.093*
	Yahoo	1.633 ± 0.405	**2.704±0.605**	2.636 ± 0.556	2.637 ± 0.556	2.644 ± 0.557	2.637 ± 0.556	2.653 ± 0.578	2.698 ± 0.553*	2.703 ± 0.553*
	LETOR	1.228 ± 0.281	1.578±0.443	1.402 ± 0.355	1.617 ± 0.411	1.748 ± 0.564	**1.834 ± 0.575**	1.400 ± 0.341	1.523 ± 0.365	1.545 ± 0.373
	LTRC	1.438 ± 0.235	1.659±0.278	1.562 ± 0.253	1.744 ± 0.281	**1.833 ± 0.341**	1.814 ± 0.310	1.567 ± 0.253	1.615 ± 0.259	1.637 ± 0.263
ExpSerendipity	Coat	0.924 ± 0.570	1.766±1.091	1.583 ± 0.895	1.862 ± 0.843	1.585 ± 0.895	1.508 ± 0.898	1.879 ± 0.993	**2.054 ± 1.122***	2.040 ± 1.117*
	Netflix	0.280 ± 0.386	0.965±0.892	1.187 ± 0.900	**1.336 ± 0.852**	0.830 ± 0.848	1.042 ± 0.952	1.290 ± 0.936	1.287 ± 0.942*	1.285 ± 0.949*
	Movielens	0.289 ± 0.422	**1.253±1.078**	1.073 ± 0.984	1.152 ± 1.053	1.201 ± 1.151	1.183 ± 1.146	1.209 ± 1.097	1.209 ± 1.097*	**1.210 ± 1.097***
	Yahoo	0.229 ± 0.300	2.754±1.380	3.858 ± 1.547	3.857 ± 1.546	3.833 ± 1.537	3.852 ± 1.540	**3.859 ± 1.563**	3.675 ± 1.659*	3.643 ± 1.668*

Table 9: ***Expnum*, *ExpDCG* and *ExpSerendipity* values with item continuation probabilities mapped to $[0.1, 0.9]$. The results with * are obtained using the BτI-H heuristic**

Metrics	Datasets	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
ExpDCG	Coat	0.293 ± 0.255	1.269±0.832	1.370 ± 0.896	**1.393 ± 0.932**	1.381 ± 0.919	1.387 ± 0.924	1.264 ± 0.881	1.269 ± 0.890*	1.317 ± 0.899*
	KuaiRec	0.299 ± 0.181	0.925±0.229	2.100 ± 0.658	**2.259 ± 0.760**	2.201 ± 0.743	2.161 ± 0.717	1.271 ± 0.436	1.399 ± 0.415*	1.453 ± 0.430*
	Netflix	0.581 ± 0.341	2.221±0.772	2.335 ± 0.674	**2.492 ± 0.777**	2.447 ± 0.765	2.481 ± 0.773	2.347 ± 0.749	2.368 ± 0.739*	2.378 ± 0.738*
	Movielens	0.525 ± 0.369	3.251±0.191	3.005 ± 0.129	**3.347 ± 0.083**	3.294 ± 0.119	3.281 ± 0.163	3.274 ± 0.157	3.274 ± 0.156*	3.275 ± 0.152*
	Yahoo	0.407 ± 0.423	2.074±1.064	1.991 ± 0.947	2.020 ± 0.944	**2.278 ± 0.995**	1.991 ± 0.947	2.051 ± 1.031	2.131 ± 0.984*	2.137 ± 0.986*
	LETOR	0.199 ± 0.150	0.304±0.219	0.377 ± 0.258	0.514 ± 0.282	0.513 ± 0.300	**0.523 ± 0.291**	0.376 ± 0.260	0.465 ± 0.253	0.472 ± 0.260
	LTRC	0.233 ± 0.220	0.433±0.359	0.667 ± 0.407	0.835 ± 0.502	0.841 ± 0.518	**0.855 ± 0.525**	0.666 ± 0.408	0.737 ± 0.433	0.751 ± 0.443
ExpSerendipity	Coat	0.124 ± 0.179	0.843±0.912	0.778 ± 0.869	0.815 ± 0.933	0.800 ± 0.890	0.774 ± 0.897	0.863 ± 0.909	0.903 ± 0.990*	**0.931 ± 1.007***
	Netflix	0.068 ± 0.164	0.715±0.790	0.836 ± 0.801	0.680 ± 0.774	0.869 ± 0.893	0.816 ± 0.889	0.871 ± 0.876	0.873 ± 0.881*	**0.876 ± 0.886***
	Movielens	0.055 ± 0.156	**1.265±1.125**	1.030 ± 0.931	0.744 ± 0.998	1.179 ± 1.134	1.164 ± 1.139	1.223 ± 1.099	1.223 ± 1.099*	1.223 ± 1.099*
	Yahoo	0.051 ± 0.143	1.842±1.588	2.461 ± 1.974	2.452 ± 1.975	2.143 ± 1.987	2.448 ± 1.954	**2.505 ± 2.043**	2.439 ± 2.076*	2.426 ± 2.078*

Table 10: Run times of proposed algorithms and baselines (sec). The results marked with * are obtained using the BτI-H heuristic

	Random	EXPLORE	DUM	MSD	MMR	DPP	B2I	B3I	B4I
Coat	0.83	2.81	0.97	1.08	1.24	0.96	1.41	4.70*	53.03*
KuaiRec	21.00	134.82	25.40	43.92	58.37	106.47	198.61	55.27*	283.62*
Netflix	10.76	180.60	14.51	14.71	14.51	50.67	181.57	72.79*	866.86*
Movielens	40.12	787.52	68.91	135.89	87.50	561.19	929.23	154.21*	1121.35*
Yahoo	53.78	1922.62	132.07	252.97	125.09	1316.29	2089.38	388.58*	3807.37*
LETOR	0.17	9.90	2.71	1.94	1.74	2.12	0.93	10.05	873.70
LTRC	0.45	28.30	6.52	4.98	4.97	5.95	2.24	42.11	3584.67