
Supplementary Material for Paper: Cold-start Recommendation by Personalized Embedding Region Elicitation

A PROOF OF THEOREM 1

We here provide the proof of Theorem 1 that are omitted in the main text.

Proof. The optimization problem to find the Chebyshev center and its radius can be rewritten as

$$\begin{aligned} \max \quad & r \\ \text{s. t.} \quad & 2(u_c + \delta)^\top (v_j - v_i) \leq \|v_j\|_2^2 - \|v_i\|_2^2 \\ & \forall \delta \in \mathcal{B}_r, \forall v_i \succsim v_j \in \mathbb{P} \\ & u_c \in \mathbb{H}, r \in \mathbb{R}_+, \end{aligned}$$

where $\mathcal{B}_r = \{\delta \in \mathbb{R}^d : \|\delta\|_2 \leq r\}$ is a d -dimensional Euclidean ball of radius r . Pick any preference $v_i \succsim v_j \in \mathbb{P}$, the semi-infinite constraint

$$2(u_c + \delta)^\top (v_j - v_i) \leq \|v_j\|_2^2 - \|v_i\|_2^2 \quad \forall \delta \in \mathcal{B}_r$$

is equivalent to the robust constraint

$$2u_c^\top (v_j - v_i) + 2 \sup_{\|\delta\|_2 \leq r} \delta^\top (v_j - v_i) \leq \|v_j\|_2^2 - \|v_i\|_2^2.$$

Because the Euclidean norm is a self-dual norm, we have

$$\sup_{\|\delta\|_2 \leq r} \delta^\top (v_j - v_i) = r \|v_j - v_i\|_2.$$

Substituting the above relationship to the optimization problem completes the proof. \square

B FURTHER EXPLANATIONS ABOUT SETTINGS AND REGION ELICITATION

In Assumption 2, the probability that an user u_0 has experienced an item v_i is given by

$$p_{0i} \triangleq w_i \times \text{sigmoid}\left(\frac{1}{c_{0i}} - \frac{\kappa_0}{\sqrt{d} - c_{0i}}\right),$$

where $c_{0i} = \|u_0 - v_i\|_2$ is the distance between the true user's and the item's embedding. In Figure 3, we visualize the dependence of p_{0i} on the parameter κ_0 . For a fixed value of the distance c_{0i} , the experience probability p_{0i} decreases monotonically in κ_0 .

Next, in a toy 2D example, we visualize the region $\mathcal{U}_{\mathbb{P}}$ in Figure 4. Initially, a new user (red star) came into our system, but we are unaware of its true embedding location. After two steps of elicitation, it is evident that the Chebyshev center moves progressively closer to the 'True User' embedding, underscoring the success of our proposed method in predicting user embeddings.

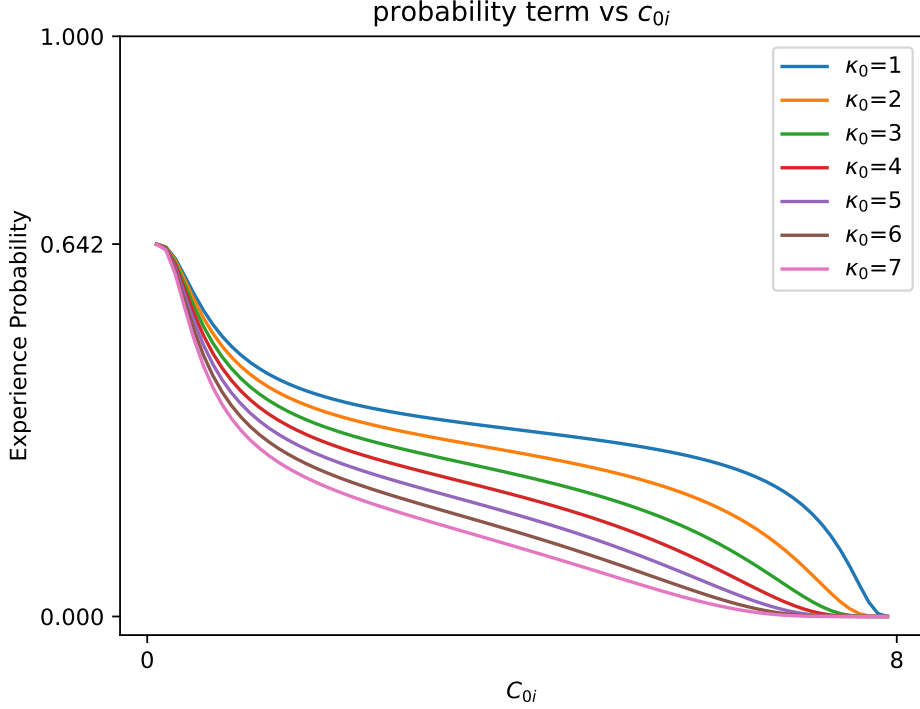


Figure 3: As the value of κ_0 increases, the probability that the user has prior experience (see Assumption 2) with an item is dampened. Plot with $d = 64$ and the maximal value of c_{0i} is $\sqrt{d} = 8$.

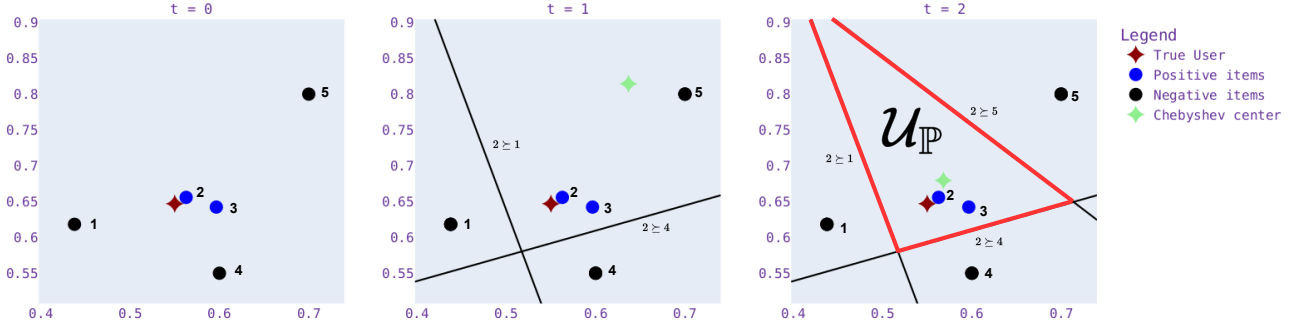


Figure 4: Illustration of our method in 2D toy example: Recall that a cut in the embedding space is created by pairing a positive item with a negative item. At time $t = 0$, when no questions have been asked, there are no cuts in the embedding space. Moving to time $t = 1$, we asked the user to elicit items 1, 2, and 4, and the user specified ‘dislike’, ‘like’, and ‘dislike’ for each respective item. This introduces two cuts in the space, and the initial Chebyshev center is calculated. Progressing to time $t = 3$, we ask the user to elicit item 5 and determine it to be a disliked item. As a result, a final cut is constructed by pairing item 2 with item 5. This process concludes with the finalization of region $\mathcal{U}_{\mathbb{P}}$

C COLD-STARTING QUERY LIST VIA DETERMINANTAL POINT PROCESSES

The main task of the “burn-in” Phase is to create a list, denoted as \mathcal{L} , comprising K popular items for querying the new user. If a user has no previous experience with an item v_i , they will indicate NA for that particular item. This NA response is uninformative because item v_i does not lead to any pair of preferences being added to the preference list \mathbb{P} as by the rule of preference construction. Therefore, when constructing the cold-start item list \mathcal{L} , it is important to consider the probability that a user has prior experience with the items. By Assumption 2, this probability is affected by two elements: the popularity of the item and the distance from the true user embedding u_0 to the item embedding v_i .

Since we do not know the user embedding u_0 , but we have information about the popularity of the items, we thus leverage this popularity information in the construction of \mathcal{L} . This line of argument also justifies the construction of the list \mathcal{L} that contains only the most popular items from the list of all possible items. To find this list \mathcal{L} , we can use a simple weighted K -medoids method: given a list of N items; the weighted K -medoids return a subset of K items to be used as cluster centers. The weighted K -medoids problem aims to minimize the total weighted squared Euclidean distance from the item embeddings to the nearest centers.

We present in this section a determinantal point process (DPP) to construct the item list \mathcal{L} . We aim to find a set of items that can balance the diversity and popularity of items oblivious to the user's true embedding. DPPs are elegant probabilistic models of global, negative correlations, and they admit efficient algorithms for sampling, marginalization, conditioning, and other inference tasks [Kulesza et al., 2012]. DPPs have been applied in various machine learning tasks, including document summarization [Perez-Beltrachini and Lapata, 2021] and image search [Chao et al., 2015]. We rely on the following L -ensemble definition of DPP.

Definition 1 (L -ensemble DPP). *Given a positive semidefinite P -by- P matrix $L \in \mathbb{S}_+^P$, an L -ensemble DPP is a distribution over all 2^P index subsets $J \subseteq \{1, \dots, P\}$ such that*

$$\text{Prob}(J) = \det(L_J) / \det(I + L),$$

where L_J denotes the $|J|$ -by- $|J|$ submatrix of L with rows and columns indexed by J .

We design the matrix L that can balance the diversity and popularity of items. We compose L as the sum of a similarity matrix S and a popularity matrix D among items:

$$L = S + D, \quad \text{where} \quad D = \text{diag}(w_i).$$

The matrix D is diagonal, and its diagonal elements capture the popularity of the items. A possible choice for the similarity matrix S is $S = V^\top V \in \mathbb{S}_+^P$ where V is the embedding matrix of the popular items. Because both S and D are positive semidefinite, the ensemble matrix L is also positive semidefinite.

We then find the combination of top- K items that fit with the construction of the cold-start querying list by solving the following problem

$$\max \{ \det(L_z) : z \in \{0, 1\}^P, \|z\|_0 = K \}, \quad (6)$$

where L_z is a submatrix of L restricted to rows and columns indexed by the one-components of z . It is well-known that the solution to problem (6) coincides with the MAP estimate of the DPP with a cardinality constraint [Kulesza et al., 2012]. Further, it is crucial to highlight that problem (6) is a submodular maximization problem since the log-probability function $\log \det(L_z)$ is a submodular function [Gillenwater et al., 2012]. Further, this problem is well-known to be NP-hard [Kulesza et al., 2012], and thus it is notoriously challenging to solve (6) to optimality. Chen et al. [2018] provides a greedy algorithm for the MAP estimation problem. The aforementioned greedy algorithm has been proven to achieve an approximation ratio of $\mathcal{O}(\frac{1}{k!})$ [Civril and Magdon-Ismail, 2009] and incur a computational complexity of $\mathcal{O}(K^2 P)$. Moreover, in order to improve the solution quality, we introduce a 2-neighborhood local search strategy. This method involves an iterative process of exchanging one element from the current set with one element from the complementary set, continuing until no additional improvement can be achieved.

D MAXIMUM LIKELIHOOD ESTIMATION OF THE TOLERANCE PARAMETER

We provide the maximum likelihood estimation for the parameters κ . Without any loss of generality, we consider a training dataset consisting of N items and M users, the user embeddings u_m and the item embeddings v_i are given. The interactions between the users and the items are presented by a binary-valued data matrix $E \in \{0, 1\}^{M \times N}$ with each E_{mi} admits values

$$E_{mi} = \begin{cases} 1 & \text{if user } m \text{ has an experience with item } i, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that there exists a global constant $\kappa \in \mathbb{R}_+$ such that E_{mi} follows a Bernoulli random variable with

$$\text{Prob}(E_{mi} = 1) = w_i \times \text{sigmoid}\left(\frac{1}{c_{mi}} - \frac{\kappa}{\sqrt{d} - c_{mi}}\right),$$

where c_{mi} is the embedding distance between the user and the item $c_{mi} = \|u_m - v_i\|_2$. Given the data matrix E and suppose that the elements E_{mi} are jointly independent, the likelihood is

$$L(\kappa|E) = \prod_{m=1}^M \prod_{i=1}^N (p_{mi}(\kappa))^{E_{mi}} (1 - p_{mi}(\kappa))^{1-E_{mi}},$$

where $p_{mi}(\kappa)$ is

$$p_{mi}(\kappa) = \frac{w_i}{1 + \exp\left(\frac{\kappa}{\sqrt{d} - c_{mi}} - \frac{1}{c_{mi}}\right)}.$$

The estimate $\hat{\kappa}_{MLE}$ minimizes the negative log-likelihood:

$$\begin{aligned} \min_{\kappa \geq 0} & \sum_{m=1}^M \sum_{i=1}^N \log \left(1 + \exp \left(\frac{\kappa}{\sqrt{d} - c_{mi}} - \frac{1}{c_{mi}} \right) \right) \\ & - \sum_{m=1}^M \sum_{i=1}^N (1 - E_{mi}) \log \left(1 + \exp \left(\frac{\kappa}{\sqrt{d} - c_{mi}} - \frac{1}{c_{mi}} \right) - w_i \right), \end{aligned}$$

which can be found by standard gradient descent algorithms.

E QUESTIONNAIRE DESIGN

Inspired by the structure of the Netflix questionnaire [Kweon et al., 2020], we devise our questionnaire methodology to capture a comprehensive set of preference pairs while minimizing user effort. Users are provided the option to skip specifying preferences, streamlining the process. In our questionnaire, users are presented with a product display, and while scrolling through, they only need to indicate ‘like’ or ‘dislike’ for products they are familiar with. An illustration of the questionnaire is provided in Figure 5. In practice, although our experimental design prompts new users to specify preferences for 100 items, our algorithm performs effectively even when utilizing an average of around 15% of user responses, evident by the user response ratio in Table 5.

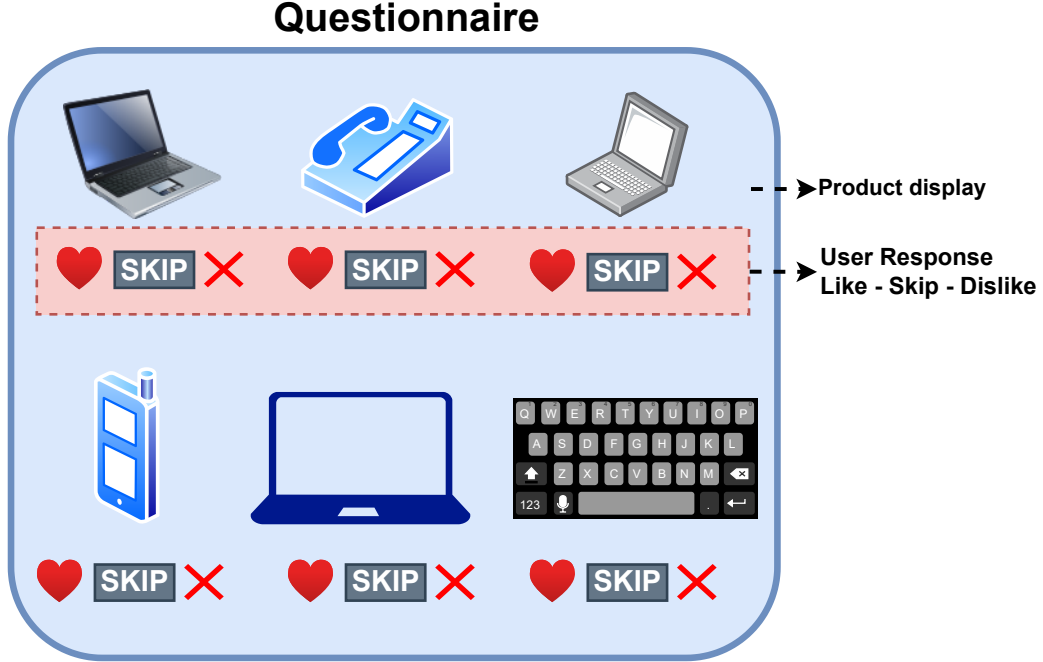


Figure 5: Illustration of our questionnaire: Taking inspiration from the Netflix questionnaire as outlined in Kweon et al. [2020], we structure each questionnaire as depicted above. Upon a new user entering our system, we prompt them to indicate their preferences for a set of items. Users can specify ‘like’ (+1), ‘dislike’ (−1), or choose to skip the item (NA).

Table 5: Number of items responded to by users using the PERE method. The response ratio is calculated over 100 queried items.

Dataset	Liked items	Disliked items	Response ratio
Amazon-Books	1.6333	14.7733	16.4066 %
Gowalla	1.0000	11.2623	12.2623 %

F ADDITIONAL NUMERICAL RESULTS

F.1 BURN-IN PHASE COMPARISON

We use LightGCN / BiVAE for the burn-in phase to generate item embedding and conduct experiments on Gowalla and Amazon-Books datasets. We employ two widely recognized and straightforward baseline methods: RMV [Fonarev et al., 2016] and K -Medoids [Liu et al., 2011]: RMV optimizes the volume of a rectangle matrix by selecting diverse yet orthogonal seed items in the embedding space. On the other hand, the K -Medoids algorithm, previously employed in a study [Liu et al., 2011], identifies representative items through cluster centroids. We slightly modify the K -medoids algorithm by considering only the items belonging to the popular items as potential centroids. Note that sequential-based preference elicitation methods, such as DPE [Parapar and Radlinski, 2021] or conditional DPP, are not applicable during the ‘burn-in’ phase. In this phase, we aim to create a standardized questionnaire for all new users entering our system. Sequential-based methods, in contrast, involve asking new questions based on the responses of previous users.

Results for the burn-in phase are summarised in Table 6. The results demonstrate that DPP (Determinantal Point Process) is the best approach for selecting initial items for the initial queries. DPP significantly outperforms baseline methods regarding performance metrics in all two datasets. The success of DPP can be attributed to its ability to effectively select a diverse set of items while considering the popularity score of each item. This combination allows DPP to balance diversity and relevance, resulting in superior performance compared to the baseline methods.

Table 6: Benchmark of performance metrics on Gowalla and Amazon-Books. Larger values are better. The best performance for any fixed number of questions is highlighted in bold. The number of items, in this case, is $K = 50$ for all methods.

Dataset	Method	HR@5	HR@10	AUC@5	AUC@10	NDCG@10	NDCG@30	MAP	MRR
Gowalla	RMV	0.1133	0.1500	0.0503	0.0839	0.0846	0.1053	0.0634	0.0636
	DRE	0.1333	0.2317	0.0532	0.1070	0.1036	0.1491	0.0499	0.0756
	k -Medoids	0.1500	0.1933	0.0669	0.1097	0.1136	0.1349	0.0859	0.0892
	DPP	0.1933	0.2433	0.0939	0.1427	0.1415	0.1734	0.1030	0.1146
Amazon-Books	RMV	0.2967	0.3633	0.2042	0.2473	0.2372	0.2590	0.1894	0.2096
	DRE	0.0483	0.1617	0.0307	0.0698	0.0674	0.1105	0.0246	0.0508
	k -Medoids	0.3200	0.3833	0.2225	0.2628	0.2624	0.2929	0.2128	0.2400
	DPP	0.3833	0.4567	0.2419	0.2968	0.3032	0.3353	0.2403	0.2730

Moreover, to show the effectiveness of our proposed sequential elicitation in Section 3, we conduct an additional experiment that compares PERE, which uses a static 50-item questionnaire in the beginning, and a series of 5 dynamic 5-item questionnaires after that, with a baseline where only a burn-in questionnaire using DPP is utilized to create a static 100-item questionnaire. Table 7 illustrates that the combination of a 50-item questionnaire along with a series of 5 dynamic 5-item questionnaires outperforms the 100-item questionnaire, which highlights the effectiveness of our PERE method.

F.2 GREEDY AND DPP COMPARISON

While the greedy method chooses the most popular item, we employ the Determinantal Point Process (DPP) in the ‘burn-in’ phase to achieve a better balance between diversity and popularity. DPP is advantageous in scenarios where preferences may diverge from mainstream popularity, ensuring a tailored and inclusive experience. Table 8 demonstrates that our method is more effective than the greedy method in constructing a personalized questionnaire for new users with 100 elicited items.

Table 7: Comparison between a burn-in questionnaire using DPP and PERE with 100 elicited items for each method on Amazon-Books dataset.

Datasets	Method	NDCG@10 \uparrow	MRR \uparrow
Gowalla	Burn-in	0.1497	0.1335
-	PERE	0.1806	0.1518
Amazon-Books	Burn-in	0.3388	0.3152
-	PERE	0.3616	0.3235

Table 8: Comparison between PERE and Greedy method on Amazon-Books dataset.

Methods	NDCG@10 \uparrow	MAP \uparrow	MRR \uparrow
Greedy	0.3415	0.198	0.3043
PERE	0.3616	0.2930	0.3235

F.3 QUESTIONNAIRE SIZE ANALYSIS

To be user-friendly, the questionnaire size should be small to avoid stressing the user’s cognitive load. We find that the number of items at each round does not significantly affect the quality of the method. What is more interesting to track is the improvement of the quality over a long history as the *total* number of questions increases. Therefore, we conduct an additional experiment to study the impact of the total number of questions on the performance metrics NDCG@10 and MRR. Table 9 shows that the performance of PERE increases with the history size.

Table 9: Performance improvements with the dynamic questionnaire size on Amazon-Books dataset.

Elicited items	5	10	20	30	40	50
NDCG@10 \uparrow	0.3111	0.3179	0.3264	0.3372	0.3564	0.3616
MRR \uparrow	0.2864	0.2905	0.2972	0.3048	0.319	0.3235

G MAIN EXPERIMENT SETTING

G.1 DATASETS DESCRIPTION

In this paper, we use Gowalla [Cho et al., 2011] dataset and Amazon-Books [Ni et al., 2019] dataset. We report the statistics of Gowalla and Amazon-Books datasets in Table 10. The description for each dataset is the following:

- Gowalla is a location-based dataset that contains information about user check-ins at various locations.
- Amazon-Books is a subset of the Amazon Product Review dataset, specifically centered on book products. This dataset comprises reviews and user ratings for various products.

Amazon-Books includes both explicit and implicit user responses related to book products, whereas Gowalla exclusively provides implicit information indicating user preferences toward different locations. We employ two well-known methods to generate collaborative filtering embeddings for items: LightGCN and biVAE. LightGCN is trained solely to predict user-item interactions, making it suitable for datasets with implicit responses. On the other hand, biVAE is designed to predict specific ratings for user-item interactions, which necessitates explicit responses. Given that Gowalla contains only implicit responses, we exclusively use LightGCN on this dataset. However, since Amazon-Books contains explicit and implicit responses, we can utilize LightGCN and biVAE on this dataset.

G.2 BASELINE DESCRIPTION

There are in total 7 baselines used in this paper, which can be divided into fixed questionnaire generation method and sequential questionnaire generation method: Fixed questionnaire generation methods:

Table 10: Characteristics of datasets used in our experiments.

Dataset	Train User #	Item #	Interaction #	Density
Gowalla	28858	40981	1027370	0.00084
Amazon-Books	51643	91599	2984108	0.00062

- RMV: Please refer to Section F.1.
- K -medoids: Please refer to Section F.1.
- DRE: initially, this method defines a categorical distribution for sampling seed items from the entire item pool. Subsequently, it simultaneously learns the categorical distributions and a neural reconstruction network to infer users’ preferences based on collaborative filtering (CF) information from the sampled seed items. Then, the encoder is utilized to select the seed items, while the decoder is used to recommend the favorite items.
- DPP: Please refer to Section C.

Sequential questionnaire generation method:

- PEO: This method presents a novel elicitation approach to construct a static preference questionnaire. It formulates the task of generating preference questionnaires, encompassing relative questions for new users as an optimization problem that can be solved in linear time of the number of items.
- Conditional DPP: Conditional DPP is a modified version of DPP that selects K items from the remaining set of items.
- DPE: This preference elicitation model employs multi-armed bandits to diversify the seed item set through topic and item diversity.

G.3 IMPLEMENTATION DETAILS

We use the standard codebase of LightGCN¹ and cornac implementation of biVAE² to generate item embedding and new user embedding. Afterward, we generate a new user according to Section 4 and use it as ground truth in our evaluation. This characteristics generation is necessary because we want to model experience probability that allows users to skip a question (NA response) in our questionnaire.

¹<https://github.com/gusye1234/LightGCN-PyTorch>

²<https://github.com/recommenders-team/recommenders/tree/main>