Heterogeneous Recommendations: What You Might Like To Read After Watching Interstellar (Technical Report)

1 Theorems and proofs

Theorem 1. *Given an item t_i, the global sensitivity of* X-Sim *is denoted by GS, and the similarity between t_i and any arbitrary item t_j is denoted by* X-Sim(t_i, t_j). *Then, the Private Replacement Selection (PRS) mechanism, which outputs t_j as the replacement with a probability proportional to $exp(\frac{\epsilon \cdot \text{X-Sim}(t_i,t_j)}{2 \cdot GS})$, provides ϵ-differential privacy.*

Proof. Consider two datasets D and D' which differ at one user, say u. We denote X-Sim (t_i,t_j) in dataset D as $q(D, t_i, t_j)$ and $I(t_i)$ as the set of items in target domain with quantified X-Sim values. The global sensitivity (GS) is defined as $max_{D,D'}||q(D, t_i, t_j) - q(D', t_i, t_j)||_1$. Our PRS mechanism outputs an item t_j as a private replacement for t_i. Then, we get the following equality:

$$\frac{Pr[PRS(t_i, I(t_i), q(D, I(t_i))) = t_j]}{Pr[PRS(t_i, I(t_i), q(D', I(t_i))) = t_j]} = \frac{exp(\frac{\epsilon \cdot q(D,t_i,t_j)}{2 \cdot GS})}{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D,t_i,t_k)}{2 \cdot GS})} \div \frac{exp(\frac{\epsilon \cdot q(D',t_i,t_j)}{2 \cdot GS})}{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D',t_i,t_k)}{2 \cdot GS})}$$

$$= \underbrace{\frac{exp(\frac{\epsilon \cdot q(D,t_i,t_j)}{2 \cdot GS})}{exp(\frac{\epsilon \cdot q(D',t_i,t_j)}{2 \cdot GS})}}_{\text{P}} \cdot \underbrace{\frac{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D',t_i,t_k)}{2 \cdot GS})}{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D,t_i,t_k)}{2 \cdot GS})}}_{\text{Q}}$$

$$P = exp(\frac{\epsilon \cdot (q(D, t_i, t_j) - q(D', t_i, t_j))}{2 \cdot GS}) \leq exp(\frac{\epsilon \cdot GS}{2 \cdot GS}) = exp(\frac{\epsilon}{2})$$

$$Q = \frac{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D',t_i,t_k)}{2 \cdot GS})}{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D,t_i,t_k)}{2 \cdot GS})} \leq \frac{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot (q(D,t_i,t_k)+GS)}{2 \cdot GS})}{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D,t_i,t_k)}{2 \cdot GS})} = \frac{exp(\frac{\epsilon}{2}) \cdot \sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D,t_i,t_k)}{2 \cdot GS})}{\sum\limits_{t_k \in I(t_i)} exp(\frac{\epsilon \cdot q(D,t_i,t_k)}{2 \cdot GS})} = exp(\frac{\epsilon}{2})$$

Therefore, we get the following inquality:

$$\frac{Pr[PRS(t_i, I(t_i), q(D, I(t_i))) = t_j]}{Pr[PRS(t_i, I(t_i), q(D', I(t_i))) = t_j]} \leq exp(\epsilon)$$

Hence, PRS provides ϵ-differential privacy. □

Theorem 2 (Similarity-based sensitivity)**.** *Given a score function $q : \mathcal{R} \rightarrow R$ and a dataset D, we define the similarity-based sensitivity corresponding to a score function $q_i(I, t_j)$ for a pair of items t_i and t_j as:*

$$SS(t_i, t_j) = max\{max_{u_x \in \mathcal{U}_{ij}}(\frac{r_{t_{xi}} \times r_{t_{xj}}}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel}), max_{u_x \in \mathcal{U}_{ij}}(\frac{r_{t_i} \cdot r_{t_j}}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} - \frac{r_{t_i} \cdot r_{t_j}}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel})\}$$

Proof. We now provide the proof of the similarity-based sensitivity. First, we define similarity-based sensitivity (SS) as follows.

$$SS(t_i, t_j) = max \parallel s(t_i, t_j) - s'(t_i, t_j) \parallel_1$$

We then insert the similarity values for $s(t_i, t_j)$. A rating vector $r_{t_i} = [r_{t_{ai}}, ..., r_{t_{xi}}, r_{t_{yi}}]$ consists of all the ratings for an item t_i. Note that here a rating $r_{t_{xi}}$ denotes the result after subtracting the average rating of user x ($\bar{r_x}$) from the actual rating provide by x for an item i. Then, we get the following equality:

$$\begin{aligned} s(t_i, t_j) - s'(t_i, t_j) &= \frac{r_{t_i} \cdot r_{t_j}}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel} - \frac{r'_{t_i} \cdot r'_{t_j}}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} \\ &= \frac{r_{t_i} \cdot r_{t_j} \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel - r'_{t_i} \cdot r'_{t_j} \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} = \frac{P}{Q} \end{aligned}$$

We assume that the profile of a user x, in D, is not present in D'. This user rated both t_i and t_j in D. Note that if this user rated one of these items or none, then the similarity value does not depend on the presence or absence of this user in the dataset. Hence, the following inequality holds: $\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel \leq \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel$.

Based on our assumption, P= $(r_{t_i} \cdot r_{t_j} \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel - r'_{t_i} \cdot r'_{t_j} \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel)$ and Q=$(\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel)$. Hence, $Q \geq 0$ and depending on whether $P \geq 0$ or $P \leq 0$ we have two conditions which are as follows.

If $P \geq 0$, then we get the following inequality:

$$\begin{aligned} \parallel s(t_i, t_j) - s'(t_i, t_j) \parallel_1 &= \frac{r_{t_i} \cdot r_{t_j} \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel - r'_{t_i} \cdot r'_{t_j} \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} \\ &\leq \frac{(r_{t_i} \cdot r_{t_j} - r'_{t_i} \cdot r'_{t_j}) \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} = \frac{(r_{t_i} \cdot r_{t_j} - r'_{t_i} \cdot r'_{t_j})}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} \end{aligned}$$

If $P \leq 0$, then we get the following inequality:

$$\begin{aligned} \parallel s(t_i, t_j) - s'(t_i, t_j) \parallel_1 &= \frac{r'_{t_i} \cdot r'_{t_j} \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel - r_{t_i} \cdot r_{t_j} \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} \\ &= \frac{(r_{t_i} \cdot r_{t_j} - r_{t_{xi}} \times r_{t_{xj}}) \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} - \frac{r_{t_i} \cdot r_{t_j} \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} \\ &= \frac{r_{t_i} \cdot r_{t_j} \times (\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel - \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel)}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} - \frac{r_{t_{xi}} \times r_{t_{xj}} \times \parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} \\ &\leq \frac{r_{t_i} \cdot r_{t_j} \times (\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel - \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel)}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel \times \parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} = \frac{r_{t_i} \cdot r_{t_j}}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} - \frac{r_{t_i} \cdot r_{t_j}}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel} \end{aligned}$$

Hence, the similarity-based sensitivity is as follows:

$$SS(t_i,t_j) = max\{max_{u_x \in \mathcal{U}_{ij}}(\frac{r_{t_{xi}} \times r_{t_{xj}}}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel}), max_{u_x \in \mathcal{U}_{ij}}(\frac{r_{t_i} \cdot r_{t_j}}{\parallel r'_{t_i} \parallel \times \parallel r'_{t_j} \parallel} - \frac{r_{t_i} \cdot r_{t_j}}{\parallel r_{t_i} \parallel \times \parallel r_{t_j} \parallel})\}$$

□

Theorem 3. *Given an item t_i, we denote its k neighbors by $N_k(t_i)$, the maximal length of all the rating vector pairs by $|v|$, the minimal similarity among the items in $N_k(t_i)$ by $Sim_k(t_i)$ and the maximal similarity-based sensitivity between t_i and other items by SS. Then, for a small constant $0 < \rho < 1$, the similarity of all the items in $N_k(t_i)$ are larger than $(Sim_k(t_i) - w)$ with a probability at least $1 - \rho$, where $w = min(Sim_k(t_i), \frac{4k \times SS}{\epsilon'} \times ln\frac{k \times (|v|-k)}{\rho})$.*

Proof. First, we recall that the probability of selecting an item is allocated by:

$$P = \frac{exp(\frac{\epsilon' \times \widehat{Sim}(t_i,t_j)}{4k \times SS(t_i,t_j)})}{\sum\limits_{l \in C_1} exp(\frac{\epsilon' \times \widehat{Sim}(t_i,t_l)}{4k \times SS(t_i,t_l)}) + \sum\limits_{l \in C_0} exp(\frac{\epsilon' \times \widehat{Sim}(t_i,t_l)}{4k \times SS(t_i,t_l)})}$$

where $\widehat{Sim(t_i,t_j)} = max(Sim(t_i,t_j), Sim_k(t_i) - w)$, $C_0 = [t_j | s(t_i,t_j) < Sim_k(t_i) - w, t_j \in I]$ and $C_1 = [t_j | s(t_i,t_j) \geq Sim_k(t_i) - w, t_j \in I]$.

We begin our proof by computing the probability of selecting a neighbor with a similarity less than $Sim_k(t_i) - w$ in each round of sampling. Given that a neighbor with similarity of $Sim_k(t_i) - w$ is still waiting for selection, the probability (p) of selecting a neighbor with similarity less than $Sim_k(t_i) - w$ is:

$$p = \frac{exp(\frac{\epsilon' \times (Sim_k(t_i) - w)}{4k \times SS})}{\sum\limits_{l \in C_1} exp(\frac{\epsilon' \times \widehat{Sim}(t_i,t_l)}{4k \times SS}) + \sum\limits_{l \in C_0} exp(\frac{\epsilon' \times \widehat{Sim}(t_i,t_l)}{4k \times SS})} \leq \frac{exp(\frac{\epsilon' \times (Sim_k(t_i) - w)}{4k \times SS})}{exp(\frac{\epsilon' \times Sim_k(t_i)}{4k \times SS})} = exp(\frac{-\epsilon' w}{4k \times SS})$$

The inequality is due to the fact that the k^{th} neighbor, with a similarity $Sim_k(t_i)$, is still present in $C_0 \cup C_1$ during the selection.

Since there are at most $|v|$ neighbors whose similarities are less than $Sim_k(t_i) - w$, the probability of choosing a neighbor with similarity s less than $Sim_k(t_i) - w$ is at most $(|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS})$. Furthermore, the probability of choosing any neighbor with similarity less than $Sim_k(t_i) - w$ leads to the following inequality after k sampling rounds:

$$P_{s < Sim_k(t_i) - w} \leq (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS}) \implies (1 - P_{s < Sim_k(t_i) - w})^k \geq [1 - (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS})]^k$$

Here, $1 - P_{s < Sim_k(t_i) - w}$ denotes the probability of selecting a neighbor with similarity at least $Sim_k(t_i) - w$. By Bernoulli's inequality, we can obtain a subsequent inquality as follows.

$$(1 - P_{s < Sim_k(t_i) - w})^k \geq 1 - k \times (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS}) \tag{1}$$

If we assume that $1 - k \times (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS}) \geq 1 - \rho$, then the inequality leads to the following inference.

$$\begin{aligned}\rho \geq k \times (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS}) &\Longrightarrow \frac{\rho}{k \times (|v| - k)} \geq exp(\frac{-\epsilon' w}{4k \times SS}) \\ &\Longrightarrow \ln(\frac{\rho}{k \times (|v| - k)}) \geq \frac{-\epsilon' w}{4k \times SS} \\ &\Longrightarrow w \geq \frac{4k \times SS}{\epsilon'} \times \ln \frac{k \times (|v| - k)}{\rho}\end{aligned}$$

Hence, we observe that with a probability of at least $1 - \rho$, the similarity of all items in $N_k(t_i)$ are larger than $Sim_k(t_i) - w$. In practice, we have to ensure that the truncated similarities satisfies the range of values i.e., $Sim_k(t_i) - w \geq 0$ for cosine or $Sim_k(t_i) - w \geq -1$ for pearson correlation, so $w = min(Sim_k(t_i), \frac{4k \times SS}{\epsilon'} \times \ln \frac{k \times (|v| - k)}{\rho})$.

□

Theorem 4. *Given an item t_i, for a small constant $0 < \rho < 1$, all items with similarities greater than $(Sim_k + w)$ are present in $N_k(t_i)$ with a probability at least $1 - \rho$ where $w = min(Sim_k(t_i), \frac{4k \times SS}{\epsilon'} \times ln \frac{k \times (|v| - k)}{\rho})$.*

Proof. Let A denote the event that a neighbor with a similarity greater than $Sim_k(t_i) + w$ has not been selected in $N_k(t_i)$ and B denote the event of selecting a neighbor with similarity less than $Sim_k(t_i)$. Then, we get the following inequality.

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \leq \frac{P(B)}{P(A)} = \frac{exp(\frac{\epsilon' \times Sim_k(t_i)}{4k \times SS})}{exp(\frac{\epsilon' \times (Sim_k(t_i) + w)}{4k \times SS})} = exp(\frac{-\epsilon' w}{4k \times SS})$$

Therefore, the probability $P_{s < Sim_k(t_i)}$ of selecting any neighbor with similarity less than $Sim_k(t_i)$ given the condition of the event A is as follows.

$$P_{s < Sim_k(t_i)} \leq (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS})$$

After the k rounds of sampling, we get the following inequality.

$$(1 - P_{s < Sim_k(t_i)})^k \geq [1 - (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS})]^k \quad \geq 1 - k \times (|v| - k) \times exp(\frac{-\epsilon' w}{4k \times SS}) \geq 1 - \rho$$

As a result, we again get the inequality: $w \geq \frac{4k \times SS}{\epsilon'} \times \ln \frac{k \times (|v| - k)}{\rho}$.

Thus, similar to the proof of the previous theorem, when $w = min(Sim_k(t_i), \frac{4k \times SS}{\epsilon'} \times ln \frac{k \times (|v| - k)}{\rho})$, all the neighbors of t_i whose similarity are greater than $Sim_k(t_i) + w$ are present in $N_k(t_i)$ with a probability at least $1 - \rho$.

□

2 Experiments

2.1 Homogeneity of X-Map

There is some inherent structural property of the data which governs when a cross-domain approach is beneficial. This structural property is dependent on the features of the items (e.g., corresponding genres or co-occurrence with other items) which can be used to partition the items into sub-domains and then leverage X-Sim. We present below one such partitioning strategy in a single-domain setting and demonstrate the application of X-Map in this setting where multiple sub-domains might exists. We consider the Movielens-20M dataset for this demonstration and partition the dataset into two sub-domains D_1 and D_2. Movielens-20M dataset consists of 20,000,263 ratings from 138,493 users for 27,278 movies with a total of 19 different genres. We partition the dataset such that the 19 genres in Movielens are divided into 10 genres in D_1 and 9 genres in D_2 as shown in Table 1. We apply X-Map on this processed dataset and then compare with a homogeneous recommendation algorithm. We use the following partition strategy to partition the single-domain into two sub-domains.

Partition strategy. The genre information of each movie is encoded into a vector $\mathbf{x} \in \mathcal{R}^{19}$ where $x_i = 1$ indicates the presence of the corresponding genre in the movie and $x_i = 0$ indicates the absence of the corresponding genre in the movie. It is important to note that each movie can have multiple genres. We next sum these vectors for all the movies to retrieve the counts of movies corresponding to each possible genre (19 genres for Movielens). We build two balanced domains by first sorting, in a descending order, the genres by the movie counts per genre and then allocating the alternate sorted genres to each sub-domain. More precisely, D_1 contains the sorted genres with even indices and D_2 contains the sorted genres with odd indices. This balancing strategy ensures that both the domains have similar fraction of popular/unpopular genres. Note that if a movie m belongs to both the domains, we add it to the sub-domain which has the most number of genres overlapping with m's set of genres and to any of the two sub-domains in case of equal overlap with both sub-domains. The original MovieLens 20M has $27,278$ movies with $20,000,263$ ratings, while the processed two sub-domains has the following statistics. Sub-domain D_1 consists of $15,119$ movies with $138,492$ users whereas sub-domain D_2 consists of $11,383$ movies with $138,483$ users. The genres distribution along with the movie counts per genre is presented in Table 1. We apply X-Map with D_1 as the source domain and D_2 as the target domain and compare with MLlib-ALS applied over the whole dataset. We observe from Table 2 that NX-Map significantly outperforms MLlib-ALS whereas X-Map, even with the additional privacy overhead, almost retains the quality of non-private MLlib-ALS.

2.2 User-based vs Item-based recommenders

Different practical deployment scenarios benefit from the proper choice of the recommendation algorithm. One requirement, which is crucial to any deployment scenario, is *Scalability*. We highlight below two factors which affect scalability in such deployment scenarios.

- Item-based recommenders leverage item-item similarities whereas user-based recommenders leverage user-user similarities. For big e-commerce players (e.g., Amazon, e-Bay), the number of items is significantly less than the number of users. Hence, such players would prefer an item-based approach for scalability purpose. For new players, the number of items would be

D_1		D_2	
Genres	**Movie counts**	**Genres**	**Movie counts**
Drama	13344	Comedy	8374
Thriller	4178	Romance	4127
Action	3520	Crime	2939
Horror	2611	Documentary	2471
Adventure	2329	Sci-Fi	1743
Mystery	1514	Fantasy	1412
War	1194	Children	1139
Musical	1036	Animation	1027
Western	676	Film-Noir	330
Other	196	–	–

Table 1: Sub-domains (D_1 and D_2) based on genres in Movielens 20M dataset.

	NX-Map	X-Map	MLlib-ALS
MAE	**0.6027**	0.6830	0.6729

Table 2: MAE comparison in a homogeneous domain setting between X-Map and MLlib-ALS on ML-20M dataset.

significantly larger than the number of users. Such new players would thus benefit from a user-based approach for scalability.

- Similarities between items tend not to vary much from day to day, or even week to week [1]. Over ranges of months, however, the similarities do vary due to various temporal factors like item popularity, behavioral drift of users. In this sense, item-item similarities are much less dynamic than user-user similarities and thus they require fewer updates.

We conducted an experiment, which we describe below, through which we demonstrate how the computation time differs for these two algorithms in two deployment scenarios. In both the scenarios, we consider the movies domain as the source domain and the books domain as the target domain.

S_1. In the first deployment scenario, we retain the original Amazon dataset. The movies dataset consists of ratings from 473,764 users for 128,402 movies whereas the books dataset consists of ratings from 725,846 users for 403,234 books. We observe that the number of users is approximately 1.8× the number of books in the target domain. This deployment scenario depicts the instance of big e-commerce players.

S_2. In the second deployment scenario, we modify the dataset of the target domain (books). The profiles of the overlapping users are retained unchanged whereas those of the non-overlapping users in the target domain are sorted, in a descending order, by the number of corresponding ratings in the profiles (profile size). Finally, only the top 100,000 users are retained in the target domain. This customized dataset consists of 104,535 users and 236,710 books in the target domain. We observe that the number of items is now nearly 2.27× the number of users. This deployment scenario depicts the instance of new e-commerce players.

Approach	S_1	S_2
	Time (s)	Time (s)
X-MAP-UB	886	**870**
X-MAP-IB	**844**	962
NX-MAP-UB	822	**805**
NX-MAP-IB	**674**	877

Table 3: Comparison between user-based (UB) and item-based (IB) recommenders in different deployment scenarios with Amazon datasets. Bold denotes faster computation time relative to the alternative.

We evaluate the recommendation quality in terms of Mean Absolute Error (MAE). We observe the following behaviour from Table 3.

- The item-based version (IB) is computationally faster than the user-based alternative (UB) in scenario S_1 where the number of users is approximately 1.8× the number of books in the target domain.
- The user-based version (UB) is computationally faster than the item-based alternative (IB) in scenario S_2 where the number of items is nearly 2.27× the number of users.

2.3 Comparison with a dimensionality reduction approach

We now compare X-MAP with a dimensionality reduction approach such as matrix factorization. For this purpose, we choose Spark's Alternating Least Squares (ALS) implementation available with its MLLIB library, denoted here by MLLIB-ALS, and apply it over the combined Amazon dataset (movies, books) of items and users while keeping the test set same as the one used for evaluating X-MAP (mentioned in the paper). We optimally tune MLLIB-ALS with varying parameters like the number of latent factors in the model (rank) or the regularization parameter (λ) to obtain the best recommendation quality. Table 4 below depicts the results of this experiment. We observe that MLLIB-ALS does not perform so well in a heterogeneous recommendation scenario which could be partially attributed to the decreased density [1] of the combined Amazon dataset (movies and books), shown in Table 5, as well as the different online behavior of the users in the two domains.

	S:Movie, T:Book	S:Book, T:Movie
NX-MAP	0.5332	0.5470
X-MAP	0.6616	0.6884
MLLIB-ALS	1.5372	1.3960

Table 4: MAE comparison between NX-Map, X-Map and dimensionality reduction approach (ALS) on Amazon datasets.

[1] Rating density is defined as the fraction of collected ratings over all the possible ratings.

Books	Movies	Books+Movies
0.0204 %	0.0569 %	**0.0147 %**

Table 5: Densities for two domains in the Amazon dataset.

References

[1] Kamal Ali and Wijnand Van Stam [*TiVo: making show recommendations using a distributed collaborative filtering architecture*]. SIGKDD, 394–401, 2004.