Heterogeneous Recommendations: What You Might Like To Read After Watching Interstellar

Response to Reviewer Comments

We thank the reviewers for their comments which we discuss below. We structure our responses by topic and refer to the reviewers as R_1, R_2 and R_3. We are also grateful to the reviewers for pointing out some typos, reference formatting as well as an incomplete dataset information [1] which we fixed in the revised version of this paper.

COMMENT 1 (R_1). *I have one main concern here: the title. Nowadays, most researchers use digital libraries and online services (such as google scholar) to find papers, and (by far) the most common searching field is the title.*

Response. We have accordingly updated the title to: *Heterogeneous Recommendations: What You Might Like To Read After Watching Interstellar.*

COMMENT 2 (R_2). *The authors implement both item-based and user-based versions of* X-MAP *and* NX-MAP. *It is not clear about the different influence of those two collaborative filtering schemes to the recommendation results. The authors should analyze and empirically evaluate these two schemes and discuss which scheme is best suitable to which situations.*

Response. We now explain the different influence of user-based and item-based collaborative filtering schemes to the recommendation results in Section 2.1 of the revised version of this paper.[2] We provide a detailed analysis of the user-based and item-based versions of X-MAP and NX-MAP in this response as well. Recall that X-MAP generates the recommendations by leveraging the AlterEgo profiles which can be treated as actual profiles in the target domain. Hence, any recommendation algorithm can leverage the generated AlterEgos for recommendation purpose.

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 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 [2]. 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 a new 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.

We evaluate the recommendation quality in terms of Mean Absolute Error (MAE). We observe from Table 1 the following behaviour.

- 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.

[1] http://snap.stanford.edu/data/web-Amazon.html

[2] The experimental results are added to our companion technical report [1] due to space constraints in our paper.

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 1: 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.

COMMENT 3 (R_2). *The authors mention the non-private version of* X-MAP *which is called* NX-MAP *for short in the paper. However, there is no comparison between* X-MAP *and* NX-MAP *in terms of prediction accuracy. They should compare their accuracy of prediction and discuss the trade-off between privacy and prediction accuracy.*

Response. We agree that the comparison between X-MAP and NX-MAP, in terms of the trade-off between privacy and prediction accuracy, was not explicitly mentioned in the previous version of the paper. We now clearly mention this trade-off in the description of Figures 6 and 7 (Section 6.3) in the revised version of this paper. Moreover, the experiments in Section 6, comparing X-MAP with the competitors, also include NX-MAP to provide an insight into the privacy overhead in X-MAP.

It is possible to observe (and compare) the accuracy achieved by X-MAP and NX-MAP from Figures 1 and 2 below (Figures 6 and 7 in the new version of the paper). In these figures, we demonstrate the effect of the privacy parameters (ϵ, ϵ') and observe that the accuracy improves with an increase in the privacy parameters. X-MAP inherently transforms to NX-MAP as the privacy parameters increase furthermore (lower privacy guarantees). Hence, the trade-off between prediction accuracy and privacy between X-MAP and NX-MAP can be observed from Figures 1 and 2 below.

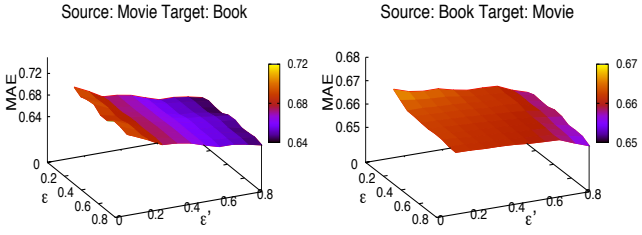

Figure 1: Privacy-quality trade-off in X-MAP-IB.

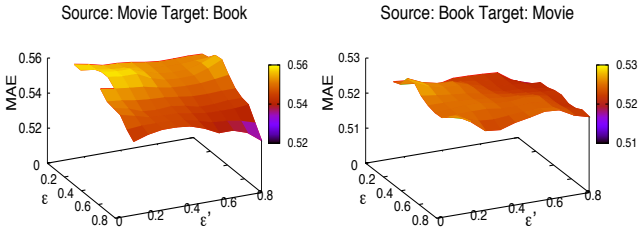

Figure 2: Privacy-quality trade-off in X-MAP-UB.

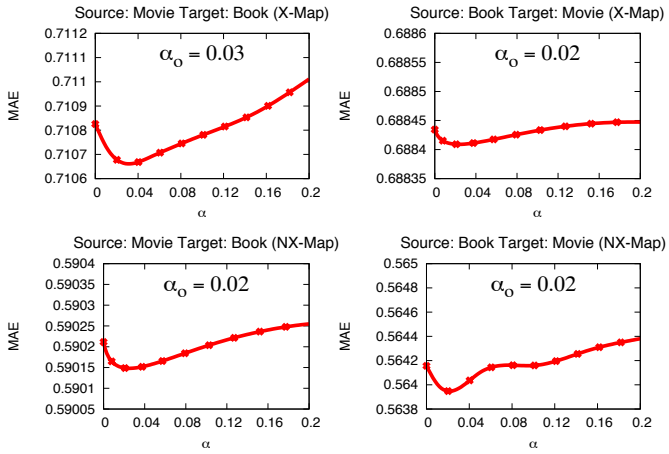

Figure 3: Temporal relevance (X-MAP, NX-MAP).

COMMENT 4 (R_2). *At the end of Section 6.2, the parameter α is said to be given in the following experiments based on Figure 5. But readers can't get the value of α either from Figure 5 or English description. Please make it clear that what's the value of α for the experiments.*

Response. We agree that the values of α (Figure 5 in the paper) were not clear in the previous version of the paper. In the revised version of the paper, we clearly mention the optimal values of the temporal parameter (α_o), in Figure 3 above (Figure 5 in our main paper), used in our experiments.

COMMENT 5 (R_2). *Please give the specific number of the overlapping users who rate both books and movies, considering those users are used in the test set.*

Response. The number of the overlapping users who rate both books and movies are indeed crucial as those users are used in the test set. The number of users in the movies dataset is 473,764 whereas the number of users in the books dataset is 725,846. The number of overlapping users in these two datasets is 78,201. We now include this detail in Section 6.1 of the revised version of the paper.

COMMENT 6 (R_2). *In Section 3.3, the authors give definition 5 directly without explaining why. What's the definition of path certainty stands for and why should path certainty be the product of those normalized weighted significance instead of sum?*

Response. The definition of *path certainty* denotes how good a path is for the similarity computations. It has been shown earlier by Ramakrishnan et al. [3] that shorter paths are better for recommendations than longer ones. In our paper, we use product instead of summation as product inherently incorporates the path length. To get an intuition, consider the following two paths (P_1 and P_2) where the fractional values on top of the arrows denote the normalized weighted significance between two consecutive items.

$$P_1\text{: A} \xrightarrow{0.1} \text{B} \xrightarrow{0.1} \text{C} \xrightarrow{0.1} \text{D}$$

$$P_2\text{: A} \xrightarrow{0.2} \text{E} \xrightarrow{0.1} \text{D}$$

We observe that a summation based technique will indicate that both the paths (P_1 and P_2) have the same significance ($C_{P_1} = C_{P_2} = 0.3$). Moreover, we can clearly see

that the path length is ignored in this case. However, if we take the product based technique, then the weighted significance would be different for the two paths ($C_{P_1} = 0.001$ and $C_{P_2} = 0.02$). This technique gives more weight to the shorter path P_2 which is also intuitive. In the revised version of the paper, we have explained the notion of path certainty along with our choice of product of the normalized weighted significance.

COMMENT 7 (R_2). *About the online deployment, I don't think all the components of* X-MAP *is executed online when an item is given for recommendation. The authors should clarify what part of the system is performed online and what part is performed offline, either in section 5 or section 6.6. Also, please show the running time of making a recommendation in the recommender component to evaluate the efficiency of the system.*

Response. Indeed, none of the components of X-MAP is executed online as that would require recomputing the AlterEgo profiles which is computationally expensive. X-MAP is periodically executed offline and the computation time for the recommendations, corresponding to all the users in the test set, is around 810 seconds on 20 nodes. Currently, we evaluate the accuracy of X-MAP in terms of MAE which is the mean of the absolute error between the predicted ratings and the actual ratings in the test set. The top-10 items (sorted by the predicted ratings), not-yet-seen by the current user, would be recommended to users in X-MAP. We have included these details in Sections 5.4 and 6.6 of the revised version of our paper.

COMMENT 8 (R_2). *The authors use* ITEMAVERAGE, REMOTEUSER, *and* ITEM-BASED-KNN *as competitors in section 6.4. Besides giving the result in Figure 8 and Figure 9, there is no analysis of why sometimes the proposed algorithms beat the comparison methods (Figure 8 (a) and Figure9 (b)) and sometimes it is beaten by the comparison methods (Figure 8 (b) and Figure 9 (a)). The authors should provide more analysis of the results.*

Response. In Figures 8 and 9 in the paper, we observe from our experiments that NX-MAP outperforms all the competitors significantly whereas X-MAP also outperforms the competitors with proper parameter tuning. We explain below the reason behind this observation which we have also updated in the revised version of the paper.

- Figure 8 (b) reveals that X-MAP performs similar to the competitors while $k \leq 50$ and then improves the quality furthermore. (Note that X-MAP provides strong privacy guarantees while the competitors are non-private.) The effect of the number of neighbors (k) is highlighted in this experiment. Unlike the competitors, a higher number of neighbors in X-MAP induces more connections among the domains (Figure 2 in the main paper) and hence enables X-MAP to explore better meta-paths between items. Moreover, better meta-paths lead to better meta-path based similarities and thereby superior recommendation quality.
- Similarly in Figure 9 (a), we observe that X-MAP performs similar to the competitors and then improve furthermore. This improvement in the quality is also attributed to the increasing connections across the domains as the number of overlapping users increase.

Furthermore, it is evident from our online deployment platform that X-SIM is highly efficient in capturing the similarities among items across different domains which is another crucial factor why NX-MAP as well as X-MAP (with privacy overhead and parameter tuning) outperforms the competitors.

COMMENT 9 (R_3). *It could be to include experiments on separating a single-domain to better understand when their technique is preferable to a simple "merge everything and treat all domains as a single, combined domain". E.g., books from different domains such as computer science text books and books about creationism are likely to be similar to "two distinct domains". Other pairs of domains might be less separated but, conceptually, one could still ask: which sci-fi book might I like to read after reading a historic novel? Put differently, there is some structural property of the data that governs when a cross-domain approach is beneficial. If all users rate plenty of items from both domains in a homogeneous manner then a cross-domain approach is unlikely to be helpful. On the other hand, even in the case of a single-domain sub-domains might exist. At the very least it would be good to evaluate their algorithm for the source=target setting to see if they simply have a better all-purpose recommendation algorithm or, more interestingly, their algorithm gives boost in a cross-domain setting but not in a single-domain setting.*

Response. We agree that though X-MAP is intended for cross-domain applications, it can be applied in a *source==target* setting. However, since there is enough auxiliary information about users' preferences in both the source and target domains (*source==target*) in such a setting, X-MAP would not improve significantly over other homogeneous recommendation algorithms if it is applied directly.

Nevertheless, as the reviewer mentioned, it is possible to apply X-MAP in a single-domain setting with some assumption on the structural property of the data to improve the homogeneous recommendation quality. 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 across the sub-domains. We present below one such partitioning strategy based on genres in a single-domain setting and demonstrate the application of X-MAP in this setting where multiple sub-domains might exist. 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 (Table 2). We apply X-MAP on this processed dataset and then compare with a homogeneous recommendation algorithm. We use the following partitioning strategy to partition the single-domain into two sub-domains.

Partitioning 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 respective movie counts per genre is presented in Table 2. We apply X-Map over this dataset, 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 3 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.

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 2: 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 3: MAE comparison in a homogeneous domain setting between NX-Map, X-Map and MLlib-ALS on ML-20M dataset.

Comment 10 (R_3). *Related to the idea of item-to-item paths is the notion of (higher order) term co-occurrence in information retrieval. Two words co-occurring in the same document is analogous to two items being rated by the same user. Terms not co-occurring directly but co-occurring with terms that co-occur is similarly analogous of longer item-user-item-user-item co-rating paths. As such, it might be interesting to explore work from the domain of information retrieval that shows that dimensionality reduction techniques improve retrieval performance because they exploit low-order co-occurrence patterns (= short paths) [1,2]. In fact, the dimensionality used for reduction does not even have to be the same for all item-item pairs [2], which is somewhat analogous to their path weighting. As such, it might be worth including a dimensionality reduction approach, such as matrix factorization, as a baseline, applied to the combined data set of items and users, rather than just to the target domain as they mention at the end of Section 4.4. (Matrix factorization is only included as a scalability baseline.)*

Response. We agree that it would be interesting to 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 [3] 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. Due to space constraints in the main paper, we add this new experiment in our companion technical report [1].

	S:Movie, T:Book	S:Book, T:Movie
NX-Map	0.5332	0.5470
X-Map	0.6616	0.6884
MLlib-ALS	0.7527	0.8237

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

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

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

1. REFERENCES

[1] X-Map GitHub repository. https://github.com/LPD-EPFL-ML/X-MAP.

[2] K. Ali and W. Van Stam. Tivo: making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 394–401. ACM, 2004.

[3] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis. Privacy risks in recommender systems. *IEEE Internet Computing*, 5(6):54, 2001.

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