Cross-Domain Recommendation for Large-Scale Data

Shaghayegh Sahebi
Department of Computer Science
University at Albany – SUNY
Albany, NY 12222
ssahebi@albany.edu

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260
peterb@pitt.edu

ABSTRACT

Cross-domain algorithms have been introduced to help improving recommendations and to alleviate cold-start problem, especially in small and sparse datasets. These algorithms work by transferring information from source domain(s) to target domain. In this paper, we study if such algorithms can be helpful for large-scale datasets. We introduce a large-scale cross-domain recommender algorithm derived from canonical correlation analysis and analyze its performance, in comparison with single and cross-domain baseline algorithms. Our experiments in both cold-start and hot-start situations show the effectiveness of the proposed approach.

KEYWORDS

Cross-Domain, Domain Selection

1 INTRODUCTION

Cross-domain recommendation systems are gradually becoming more attractive as a practical approach to improve quality of recommendations. Number of social systems that collect user interaction and preferences in different domains is constantly increasing. Accordingly, using information contributed by users in one system to help generate better recommendations in another system in a related domain has become more and more valuable. Especially important in this context is the ability of cross-domain collaborative filtering to soften the cold-start situation by offering meaningful suggestions at the very start of user interaction with a new domain. Starting with a few proof-of-concept studies [1, 2, 5, 15], cross-domain recommenders emerged in a sizable stream of research in the recommender systems field.

Yet, in some sense, the work is still in early stages. While many different models have been proposed and explored, the dominating approach to exploring new cross-domain recommendation ideas is to use public datasets that are relatively small in comparison with the full scale of data (items and users) in real-life recommender

systems. Full-scale cross-domain datasets are hard to find, so authors frequently use simulated cross-domain datasets. For example, Iwata and Takeuchi propose a matrix factorization based approach in [7] where neither users nor items are shared between domains. Although they used a large-scale dataset (using EachMovie, Netflix, and MovieLens), their large-scale dataset is not from a cross-domain system. Rather, this movie rating dataset is divided into random user and item splits. A similar splitting in large-scale movies domain can be seen in [11]. Moreover, the rare large-scale cross-domain experiment reports in literature focus mostly on content-based cross-domain recommenders [3, 9, 14]. In [8], Loni et al. use factorization machines for domains in a large-scale Amazon dataset. In their experiments, better use of within domain information generated better results compared to using cross-domain information. While the current literature show the importance of cross-domain recommender systems, the limitations reviewed above do not allow us to see how cross-domain recommender algorithms scale up.

This paper attempts to fill in the gap of design and evaluation of large-scale cross-domain recommenders by proposing a cross-domain collaborative filtering algorithm and evaluating it using a dataset collected from a multi-domain recommender system, Imhonet. The proposed algorithm, CD-LCCA, is specifically designed for scalability.

The proposed approach relies on canonical correlation analysis (CCA) [6] for transferring information from source domain to target domain. CCA has been used in context-aware single-domain recommendation [4], content-based cross-domain recommendation [3], and medium-scale cross-domain collaborative filtering [13]. However, it has not been scaled for large-scale cross-domain collaborative filtering. In this paper, we use a computationally efficient implementation of CCA to model cross-domain recommendations in a large-scale dataset. We present our model in Section 2. We compare the performance of our model with cross-domain and single domain baselines in Section 3, and analyze its cold-start behavior in Section 4. Finally, we present a time performance analysis of the algorithm in Section 5.

2 LARGE-SCALE CCA-BASED CROSS-DOMAIN ALGORITHM (CD-LCCA)

2.1 Background

CCA is a multivariate statistical model that studies the interrelationships among sets of multiple dependent variables and multiple independent variables[6]. Calculating CCA can be very resource-consuming especially in traditional approaches that should calculate QR-decomposition or singular value decomposition of large data matrices. To avoid this problem, Lu and Foster developed an iterative algorithm that can approximate CCA on very large

datasets[10]. This approach relies on LING, a gradient-based least squares algorithm that can work on large-scale matrices. To compute CCA in L-CCA in [10], first a projection of one of the data matrices on a randomly-generated small matrix is produced. Then, a QR-decomposition of this smaller matrix is calculated. After that, CCA is calculated iteratively, by applying LING on the reduced-sized QR-decompositions of the original data matrices, in each iteration. Every time after running LING, a QR-decomposition is calculated for numerical stability. Here, we build our large-scale cross-domain recommender algorithm based on L-CCA proposed by Lu and Foster.

2.2 Model

Large scale CCA finds a lower-dimensional representation of each of the input matrices and then calculates the canonical correlation analysis between these two smaller matrices. To base our cross-domain recommender algorithm on LCCA, suppose that we have a $n \times m$ source domain rating matrix X and a $n \times p$ target domain rating matrix Y. Here, n represents number of shared users between source and target domains; m shows number of items in source domain; and p shows number of items in target domain. The goal of our model is to estimate user ratings in the target domain (Y_{ij}s), given user ratings in the source domain (X_{ij}s). We find the mapping that is between these two domains using LCCA as explained in the following.

Suppose that X_c ($n \times x_c$) is a lower dimensional matrix that represents source domain rating matrix X, and Y_c ($n \times y_c$) is a lower dimensional matrix that represents target rating matrix Y in the LCCA algorithm. Calculating the canonical correlations between X_c and Y_c leads us to two canonical variates ($X_c W_{x_c}$ ($n \times k_{cca}$) and $Y_c W_{y_c}$ ($n \times k_{cca}$)) and a diagonal matrix P ($k_{cca} \times k_{cca}$) that shows the canonical correlation between these variates. Using these canonical correlations and variates, we can map X_c to Y_c (and vice versa). For example, Y_c can be achieved using Equation 1.

$$Y_c = X_c W_{x_c} P W_{y_c}^T \tag{1}$$

Although Equation 1 relates the lower dimensional representations of original source and target domains (X_c and Y_c), we need to map the original source and target matrices (X and Y) to estimate user ratings in them. To build a relationship between original source and target domain matrices, we first look at the relationship between each domain matrix and its lower dimensional representation. Without loss of generality, we consider the source domain relationships. X_c is built in the first step of LCCA by solving an iterative least square problem, having a QR-decomposition in each iteration. Although we loose the mapping information between X and X_c in this iterative process, having both X and final X_c matrices, we can restore their mapping. We can rewrite X and X_c's relationship as in $X_c = XM$. Here, M is a $m \times c_x$ mapping that projects X into X_c; and thus:

$$M = X^{-1} X_c \tag{2}$$

The same can be applied to find the mapping of target rating matrices Y and its lower-dimensional representation Y_c (Equation 3).

$$N = Y^{-1} Y_c \tag{3}$$

Combining Equations 1, 2, and 3, we can now map between the original source and target rating matrices as presented in Equation 4 and have an estimation of user ratings in the target domain ($\hat{Y}$).

$$\hat{Y} = XMW_{x_c} P W_{y_c}^{-1} N^{-1} \tag{4}$$

When the rating matrix sizes are too large, calculating the multiplications in 4 can be resource-consuming. To resolve this, we take advantage of the fact that $[A|B]^{-1}C = [A^{-1}C|B^{-1}C]$, and separate the source matrix into multiple smaller matrices, using column-wise partitioning. Then, we perform the multiplication on each of these matrices and eventually join the results together.

Equation 4 gives us the opportunity to relate the source and target domain rating matrices. Based on that, we can estimate the ratings in target domain Y based on ratings in source domain X. In other words, we can estimate user i's rating on item j from the target domain, given user i's ratings in the sources domain using Equation 5. Thus, our cross-domain recommender system can suggest the most relevant items to users in the target domain, having user ratings in the source domain. In the following sections, we evaluate our proposed model both in the cold-start and hot-start setting, using a large-scale dataset

$$\hat{y}_{i,j} = \Sigma_{q=1}^{m} X_{i,q} \Sigma_{o=1}^{c_x} M_{q,o} \Sigma_{l=1}^{k_{cca}} W_{x_{c_o,l}} P_{l,l} \Sigma_{r=1}^{c_y} W_{y_{c_l,r}} N_{r,j}^{-1} \tag{5}$$

3 DO LARGE-SCALE CROSS-DOMAIN ALGORITHMS HELP?

In our first set of experiments we study if the proposed cross-domain recommender system is useful in large-scale datasets. In other words, by comparing the cross-domain and single-domain recommendation results, we explore if target domain user data can be enough for achieving good recommendations in large-scale datasets; or if auxiliary information can be helpful.

3.1 Dataset

We use the Imhonet dataset for carrying our experiments in this paper. This is an anonymized dataset obtained from an online Russian recommender service *Imhonet.ru*. It allows users to rate and review a range of items from various domains, from books and movies to mobile phones and architectural monuments. Imhonet is a true multi-domain system: while it supported different domains, each domain was treated almost as an independent sub-site with separate within-domain recommendations. This system also contains many aspects of a social network, including friendship links, blogs and comments. Combination of explicit user feedback (ratings) and diverse domains makes Imhonet very unique and valuable for cross-domain recommendation. We use a dataset that includes Imhonet's four large domains - books, movies, games, and perfumes. It contains a full set of user ratings (at the time of collection) across four domains.Each rating record in the dataset includes a user ID, an item ID, and a rating value between zero (not rated) and ten. The same user ID indicates the same user across the sets of ratings. Some basic statistics about this dataset are shown in Table 1. To pre-process this dataset we find shared users across category pairs.

Table 1: Basic Statistics for Imhonet Dataset.

	Book	Game	Movie	Perfume
user size	362448	72307	426897	19717
item size	167384	12768	90793	3640
density	2.22E-04	0.0014	7.30E-04	0.0035
record #	13438520	1324945	28281946	253948
average rating # per user	37.0771	18.2339	6.63E+03	12.8796
average rating # per item	80.2856	103.7708	311.4992	69.7659

3.2 Experiment Setup

To run the experiments, we used a user-stratified 5-fold cross-validation setting: 20% of users are selected as test users and the rest of them (80%) are selected as training users. We recommend items to test users given the training data and 20% of their ratings.

Some of the algorithms have parameters that should be selected by cross-validation. To find the best set of parameters for each algorithm, we select 15% of users as "validation" users and remove 80% of their ratings from the training set. We repeat the experiments 5 times, and report the average performance of algorithms. To measure the performance of algorithms, we use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

For the single-domain algorithm, we use only the target domain dataset. However, for cross-domain algorithms, we have both source and target datasets. To be able to compare single and cross-domain algorithms, we remove the same set of ratings for all of the algorithms.

Table 2: Correlation algorithms' RMSE with each other. *: significant with p-value < 0.01**.**

	CD-LCCA	CD-SVD	SD-SVD
CD-LCCA	1	0.1993	-0.1909
CD-SVD	0.1993	1	0.7416*
SD-SVD	-0.1909	0.7416*	1

3.3 Results

There are four domains in the dataset: books, movies, perfumes, and games. This results in having 12 domain pairs to study. Some of the statistics of domain pairs are presented in Table 3. We can see that "books" and "movies" domain pairs have the most number of users and "games" and "perfumes" domains have the least number of common users. The "books" domain has the maximum and "perfumes" domain has the minimum number of items. Also, the "books" domain is among the most sparse domains, while the "perfumes" domain is the least sparse one.

We run the proposed and baseline algorithms on each of these domain pairs. Figures 1 and 2 show RMSE and MAE of algorithms on 12 domain pairs of Imhonet. The reported errorbars represent a 95% confidence interval for errors. As we can see in these figures, the use of cross-domain data with a competitive algoritms originally designed for a single domain doesn't produce a significant impact: the single-domain algorithm (SD-SVD) performs better than, or similar to, cross-domain baseline (CD-SVD) in many domains. Only in "movie → book" and "game → movie" domain pairs, CD-SVD is
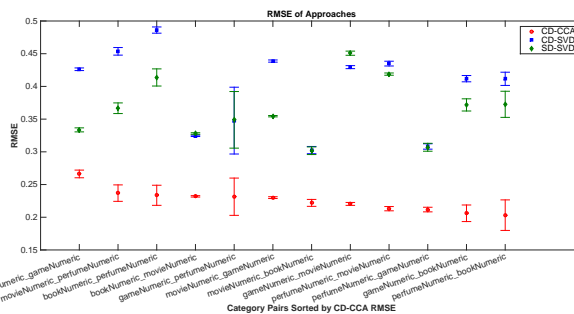

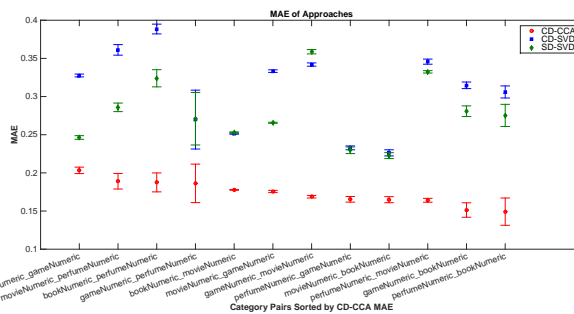**Figure 1: RMSE of algorithms on** 12 **Imhonet domain pairs**

Figure 2: MAE of algorithms on 12 **Imhonet domain pairs ordered by the MAE of the CD-LCCA**

significantly better than SD-SVD. The domains in these two pairs are semantically closer, compared to other domain pairs. However, CD-LCCA performs significantly better than both CD-SVD and SD-SVD in all of the domain pairs. Thus, CD-LCCA is able to see beyond the semantic relationships between domains and capture their latent similarities that may not seem intuitive. Also, we can see that confidence intervals in most of the domain pairs (except for "game → perfume" and "perfume → book") are small.

To understand if average error of algorithms are related to each other in different domain pairs, we look at RMSE correlations between algorithms that are reported in Table 2. Here, we see that RMSE of CD-SVD and SD-SVD algorithms are highly correlated with each other. However, CD-LCCA's RMSE does not have any significant correlations with the two baseline algorithms' performance.

Altogether, we conclude that CD-LCCA is helpful in estimating user preferences using auxiliary domain information in large-scale datasets; the baseline cross-domain algorithm that is not designed for this purpose (CD-SVD) may harm the recommendation results rather than helping; error of baseline recommender algorithms are correlated; and CD-LCCA can understand unintuitive, but useful, similarities between domain pairs that are not discovered by CD-SVD.

Table 3: Domain and domain-pair data size statistics for the Imhonet dataset

| source	target	user size	source item size	target item size	source density	target density
book	game	41756	125688	11407	0.0007	0.0020
book	movie	186877	155765	85892	0.0003	0.0014
book	perfume	16750	105805	3545	0.0011	0.0037
game	book	41756	11407	125688	0.0020	0.0007
game	movie	49784	11715	75599	0.0019	0.0028
game	perfume	6297	6854	3232	0.0030	0.0041
movie	book	186877	85892	155765	0.0014	0.0003
movie	game	49784	75599	11715	0.0028	0.0019
movie	perfume	17882	63708	3565	0.0041	0.0037
perfume	book	16750	3545	105805	0.0037	0.0011
perfume	game	6297	3232	6854	0.0041	0.0030
perfume	movie	17882	3565	63708	0.0037	0.0041
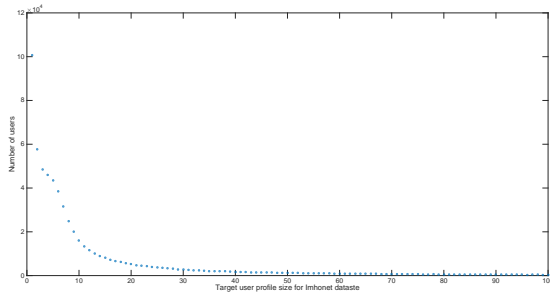
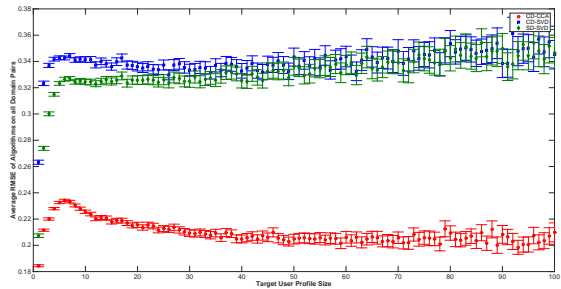**Figure 3: Target profile sizes of users in Imhonet dataset**

Figure 4: User-based RMSE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size
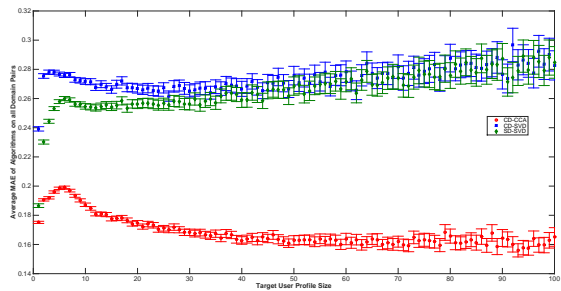

Figure 5: User-based MAE of algorithms in the Imhonet dataset, averaged on all domain-pairs and sorted based on the users' target domain profile size

4 DO LARGE-SCALE CROSS-DOMAIN ALGORITHMS ALLEVIATE COLD-START?

One of the major problems in recommender systems literature is the cold-start problem [12]. Cross-domain recommenders aim to alleviate this problem by transferring target user's source profile information for recommendation in target domain. In CD-LCCA, this transfer happens by mapping source and target domains using canonical variates and correlations as in Equation 4. In this section, we investigate the success of such transfer by comparing CD-LCCA, CD-SVD, and SD-SVD's performance in cold-start setting. To understand how each of these algorithms perform in cold-start setting, we group test users of each dataset based on their target domain profile size. Then, we calculate the error for each group of these users in each of the algorithms. Figure 3 shows number of test users vs. target domain profile sizes in all of the domain pairs. We can see that most of the test users have a small profile size (less than 10 items) in the target domain. There are a few users with 100 and more items in their target profile. To have a better plot, we skipped showing these users. Also, there is a concave shape at small (less than 10) target domain profile sizes. This happens because Imhonet has asked some users to rate at least 20 items, for providing recommendations to them. Since we only use 20% of test user ratings in their target profiles, this increase in the profile size happens for the profiles that have less than 10 items.

Figures 4 and 5 show the RMSE and MAE of algorithms in the cold-start setting based on target user profile size, averaged for all of the domain pairs. As we can see in these pictures, in average on all domain-pairs, CD-CCA performs significantly better than both of the baselines. Also, the single-domain baseline (SD-SVD) in average performs better than the cross-domain baseline (CD-SVD). In smaller profile sizes SD-SVD's error is significantly lower than

CD-SVD's error. As the target domain profile size grows, the errors of two baseline algorithms have no significant differences.

To have a better understanding of cold-start situation in each of the domain pairs, we look at the results of domain-pair combinations separately. Figures 6 and 7 show each algorithm's cold-start RMSE and MAE in each of the domain pairs. Note that we have plotted the errors for target profile sizes ranging from one to 100 items. But, in some domain pairs (e.g., "game → "perfume"), maximum user profile size is less than 100 and thus the plot is discontinued.

As we can see, for small profile sizes, in all domain pairs except "game → perfume" CD-LCCA performs significantly better than baseline algorithms. This shows that CD-LCCA can successfully transfer useful information from most source domains to target domain, especially in cold-start situation. For "book" and "movie" target domains the superior performance of CD-LCCA continues in large profile sizes. But, in "game" and "perfume" target domains performance difference of algorithms is insignificant after users have enough items in their target profile (between 25 and 45 items for different domain pairs). There are fewer users with larger profile sizes in these domains. Thus, we have lower confidence in algorithms' performance and wider confidence intervals, leading to insignificant differences.

Comparing CD-SVD and SD-SVD, we can see that they mostly have similar results. In all experiments with "movie" domain as the source domain, SD-SVD performs significantly better than CD-SVD from the beginning. But in "game → movie" and "perfume → movie", CD-SVD can be significantly better than SD-SVD especially in larger profile sizes. Accordingly, in smaller target profile sizes not only CD-SVD does not help, but also it can harm recommendation results. This shows that while CD-LCCA can efficiently use the extra source domain information, CD-SVD cannot handle this information effectively.

Looking at error trends, for some domain pairs (e.g., "movie → book" and "game → movie"), we see an initial error increase as the target profile size grows. Although we expect to see smaller errors, as we have more information from users in target domain, the observed trend is against such expectation. This trend happens in all algorithms including the single-domain baseline (SD-SVD). Thus, such behavior cannot be attributed to using extra information in cross-domain algorithms.

Altogether, we can conclude that not only CD-LCCA can handle extra information from the semantically-related target domain efficiently, but it also can understand the relationship between source and target domains that appear to be unrelated.

5 PERFORMANCE ANALYSIS

In CD-LCCA, calculating large-scale CCA costs $O(Nnp(N_2 + k_{pc}) + Nnk^2)$, in which N is number of iterations for least squares; n is number of data points (users); p is the number of items in the target domain; N_2 is the number of iterations to compute Y_r using gradient descent; k_{pc} is the number of top singular vectors used in LING; and k is the number of components. The multiplications in CD-LCCA depend on the number of nonzero elements in matrices. In the worst case of multiplying dense matrices, the multiplications cost $O(npk + nk^2)$. Thus, as a whole, CD-LCCA costs $O(Nnp(N_2 + k_{pc}) + Nnk^2 + npk)$. Since $k_{pc} \leqslant k$ and $k \leqslant p$, CD-LCCA costs less than $O(Nnp(N_2 + k))$.

In our experiments, we ran all of the algorithms on two similar machines: a MacOS machine with 64GB RAM and two 4-core Intel Xeon, 2.26GHz CPUs and a Linux machine (CentOS) with 64GB RAM and two 4-core Intel Xeon, 2.40GHz CPUs. On average, running CD-LCCA in Matlab on each domain pair took 21210 seconds (close to 6 hours), while running CD-SVD with GraphChi took almost 4 hours.

6 CONCLUSIONS

This work presented a large-scale cross-domain collaborative filtering approach, CD-LCCA. Our experiments on a large-scale user-item rating dataset with 12 domain pairs showed that cross-domain collaborative filtering can be helpful even in large-scale target domains. We saw that CD-LCCA improves recommendation results in both hot and cold-start settings in all domain pairs. But, the baseline cross-domain algorithm helped only in domain pairs with higher semantic similarities. In some cases, adding auxiliary information in the baseline cross-domain algorithm harmed the results. Thus, we concluded that CD-LCCA is able to capture unintuitive relationships between different domains, that are not being understood by the baseline algorithms. Our cold-start analysis showed that the proposed model is especially helpful in the cold-start setting.

REFERENCES

[1] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-Domain Mediation in Collaborative Filtering. In *Proceedings of the 11th international conference on User Modeling (UM '07)*. Springer-Verlag, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-540-73078-1_44

[2] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2008. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction* 18, 3 (Aug. 2008). DOI: https://doi.org/10.1007/s11257-007-9042-9

[3] Ali Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *International World Wide Web Conference Committee (IW3C2)*.

[4] Siamak Faridani. 2011. Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In *Proceedings of the fifth ACM conference on Recommender systems (RecSys '11)*. ACM, New York, NY, USA. DOI: https://doi.org/10.1145/2043932.2044005

[5] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*.

[6] Harold Hotelling. 1936. Relations Between Two Sets of Variates. *Biometrika* 28, 3/4 (1936). DOI: https://doi.org/10.2307/2333955

[7] Tomoharu Iwata and Koh Takeuchi. 2015. Cross-domain recommendation without shared users or items by sharing latent vector distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. 379–387.

[8] Babak Loni, Alan Said, Martha Larson, and Alan Hanjalic. 2014. 'Free lunch'enhancement for collaborative filtering with factorization machines. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 281–284.

[9] Yucheng Low, Deepak Agarwal, and Alexander J Smola. 2011. Multiple domain user personalization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 123–131.

[10] Yichao Lu and Dean P Foster. 2014. large scale canonical correlation analysis with iterative least squares. In *Advances in Neural Information Processing Systems*. 91–99.

[11] Weike Pan, Evan Wei Xiang, and Qiang Yang. 2012. Transfer Learning in Collaborative Filtering with Uncertain Ratings.. In *AAAI*.

[12] Denis Parra and Shaghayegh Sahebi. 2013. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2*. Springer, 149–175.

[13] Shaghayegh Sahebi and Peter Brusilovsky. 2015. It Takes Two to Tango: An Exploration of Domain Pairs for Cross-Domain Collaborative Filtering. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 131–138.
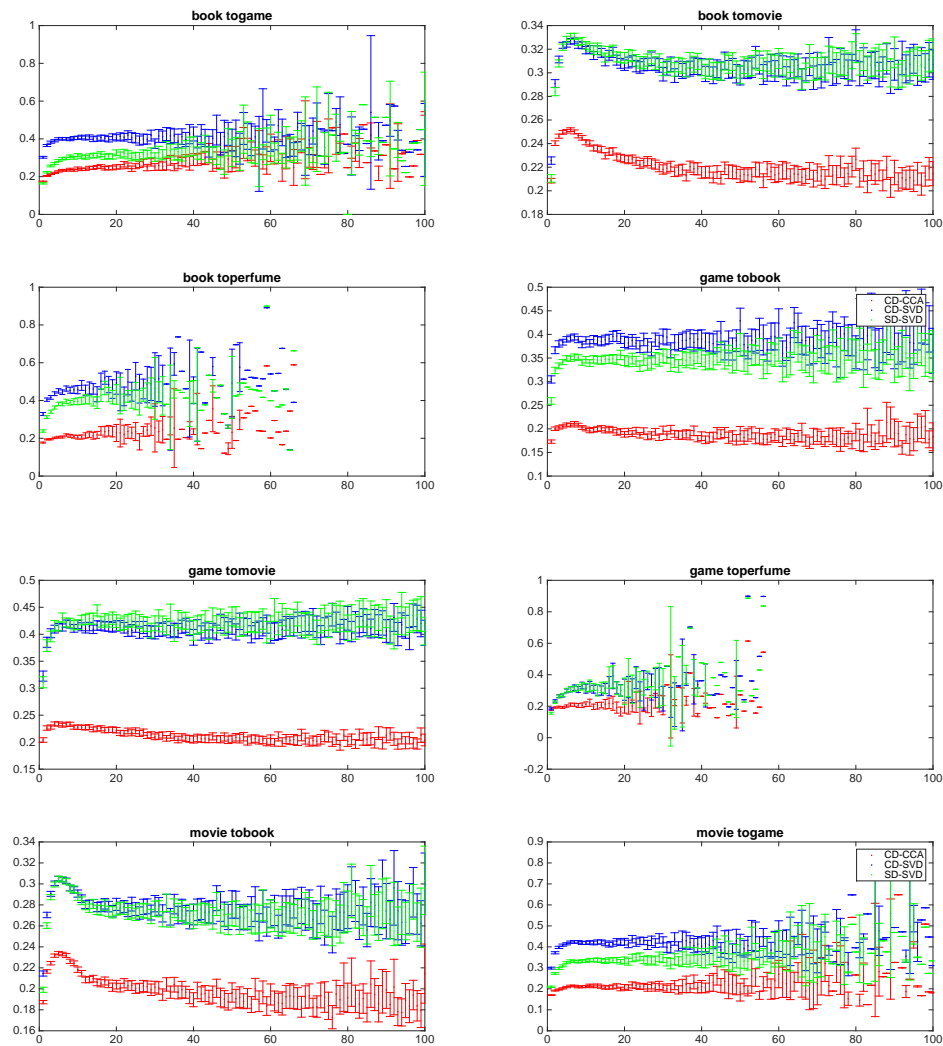
Figure 6: User-based RMSE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size

[14] Weiqing Wang, Zhenyu Chen, Jia Liu, Qi Qi, and Zhihong Zhao. 2012. User-based collaborative filtering on cross domain by tag transfer learning. In *Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining*. ACM, 10–17.

[15] Pinata Winoto and Tiffany Tang. 2008. If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations. *New Generation Computing* 26 (2008). Issue 3. http://dx.doi.org/10.1007/s00354-008-0041-0 10.1007/s00354-008-0041-0.
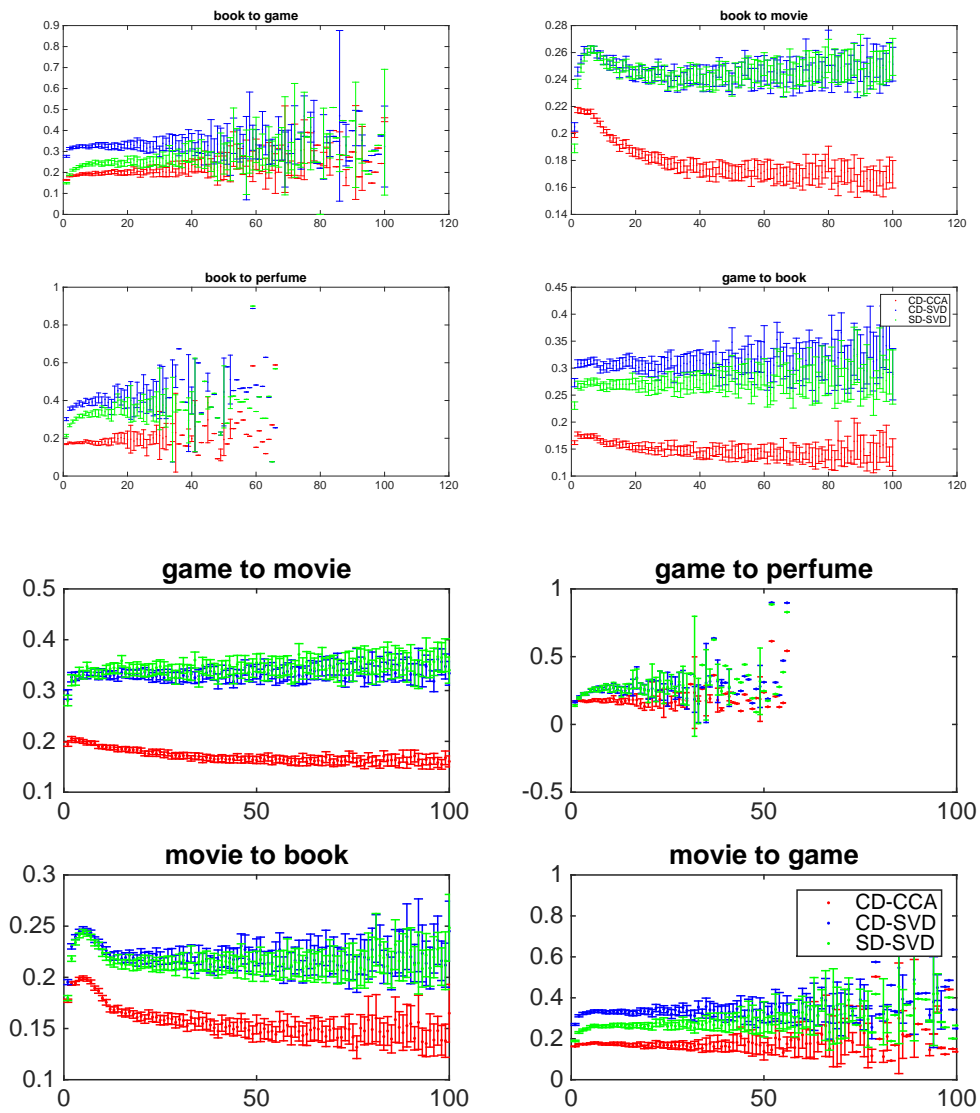
Figure 7: User-based MAE of algorithms in Imhonet dataset, averaged on each domain-pair and sorted based on the users' target domain profile size