Cross-domain Recommendations without Overlapping Data: Myth or Reality?

Paolo Cremonesi
Politecnico di Milano
Italy
paolo.cremonesi@polimi.it

Massimo Quadrana
Politecnico di Milano
Italy
massimo.quadrana@polimi.it

ABSTRACT

Cross-domain recommender systems adopt different techniques to transfer learning from source domain to target domain in order to alleviate the sparsity problem and improve accuracy of recommendations. Traditional techniques require the two domains to be linked by shared characteristics associated to either users or items. In collaborative filtering (CF) this happens when the two domains have *overlapping* users or item (at least partially). Recently, Li et al. [7] introduced *codebook transfer* (CBT), a cross-domain CF technique based on co-clustering, and presented experimental results showing that CBT is able to transfer knowledge between non-overlapping domains. In this paper, we disprove these results and show that CBT does not transfer knowledge when source and target domains do not overlap.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms and theory

Keywords

Accuracy measures; Cold start; Matrix factorization; Ratings aggregation; Sparsity

1. INTRODUCTION

In 2009, Li et al. [7] proposed *codebook-transfer* (CBT), a cross-domain collaborative filtering (CF) recommender system based on co-clustering and tri-matrix factorization. The core of CBT is based on a two-step process: extraction of knowledge (the *codebook*) from the source domain and injection of knowledge in the target domain. Both steps are performed by using a constrained tri-matrix factorization. The injection phase is used to reduce the sparsity of the target domain. After the injection, traditional CF algorithms are trained on the target domain.

The Li et al. paper [7] claims that CBT is able to transfer learning from the source domain to the target domain even in the case of totally disjoint domains, i.e., domains without shared users or items. They prove this by comparing the accuracy of a traditional CF algorithm on the target domain before and after the injection of the codebook from the source domain. The algorithm adopted for the comparison is a user-based k-nearest-neighbors (kNN) collaborative filtering based on Pearson correlation. The comparison shows a significant increase of accuracy after the injection.

However, in this paper we provide an alternative explanation to the increase of accuracy experimented with CBT that does not involve transfer of knowledge between domains. The contributions of this paper are three-fold.

- First, we provide empirical evidence that CBT improves the accuracy of CF without transferring knowledge from source to target domain[1].
- Second, we show that the injection of the codebook in the target domain is equivalent to a two-matrix factorization algorithm without transfer of knowledge from the source domain.
- Third, as a consequence of this, we show that the increase of accuracy measured in [7] is due to a pitfall in the evaluation procedure as it compares a user-based kNN algorithm (before the injection) with a matrix-factorization algorithm (after the injection) and matrix-factorization is known to outperform kNN.

The rest of the paper is organized as follows. In Section 2, we introduce the cross-domain CF problem and describe the CBT algorithm. In Section 3, we show that the CBT algorithm does not transfer knowledge. Finally, the conclusions are given in Section 4.

2. CROSS-DOMAIN CF

Cross-domain CF systems aim to generate or enhance recommendations in a target domain by exploiting knowledge (mainly user preferences) from source domains. One of the motivating use cases of cross-domain recommendations is the well known cold-start problem, which prevents a recommender system from generating recommendations due to

[1]Code and datasets are available for download from; http://home.deib.polimi.it/cremones/recsys/cbt.zip

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
Copyright 2014 ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645769.

the lack of sufficient information about a user or item. In a cross-domain setting, a recommender system may draw on information acquired from other domains to alleviate such problem. For instance, a user's preferred movie genres are likely to be derived accurately from her/his preferred literature genres. This example is based on the assumption that there are similarities or correspondences either between the user preferences or between the items in the source and target domains. Cross-domain recommender systems leverage these dependencies through considering, for example, overlaps between the observed user or item sets, similarities of item attributes, features of rated items, or correlations between user preferences in the domains. Then, they apply a variety of techniques for enriching the knowledge possessed by the target domain, and improving the quality of recommendations.

The research on cross-domain recommendation has generally aimed to exploit knowledge from a source domain $\mathcal{D}_S$, to perform or improve recommendations in a target domain $\mathcal{D}_T$. Respectively for such domains, let $\mathcal{U}_S$ and $\mathcal{U}_T$ be their sets of users, and let $\mathcal{I}_S$ and $\mathcal{I}_T$ be their sets of items. The users of a domain are those that have some ratings for items of the domain. In order to alleviate cold-start and sparsity problems in the target domain, cross-domain systems recommend items in the target domain by exploiting knowledge from both source and target domains, i.e., recommend items in $\mathcal{I}_T$ to users in $\mathcal{U}_S$ by exploiting knowledge about $\mathcal{U}_S \cup \mathcal{U}_T$ and/or $\mathcal{I}_S \cup \mathcal{I}_T$. In general, to perform this type of recommendation, some data relations or overlaps between domains are needed, and approaches aim to establish explicit or implicit knowledge-based links between the domains.

As discussed by Fernández-Tobías et al. [4], domains can be linked by means characteristics associated to users or items. Following the notation used in [4], let $\mathcal{X}^{\mathcal{U}}$ and $\mathcal{X}^{\mathcal{I}}$ be the sets of characteristics used to represent the users and items, respectively. The nature of these characteristics may be diverse, e.g. ratings, user demographics, item attributes. Two domains $\mathcal{D}_S$ and $\mathcal{D}_T$ are linked if $\mathcal{X}_S^{\mathcal{U}} \cap \mathcal{X}_T^{\mathcal{U}} \neq \varnothing$ or $\mathcal{X}_S^{\mathcal{I}} \cap \mathcal{X}_T^{\mathcal{I}} \neq \varnothing$, i.e., if there are user or item characteristics shared by the domains.

In collaborative filtering systems, the user preferences are usually modeled as a matrix $R \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, in which an element r_{ui} is the rating assigned by user u to item i. Users are described through the ratings they assign to items. Thus, $\mathcal{X}_S^{\mathcal{U}} = \mathcal{I}_S$ and $\mathcal{X}_T^{\mathcal{U}} = \mathcal{I}_T$. Similarly, items are described in terms of the ratings received by users, i.e., $\mathcal{X}_S^{\mathcal{I}} = \mathcal{U}_S$ and $\mathcal{X}_T^{\mathcal{I}} = \mathcal{U}_T$. Domains $\mathcal{D}_S$ and $\mathcal{D}_T$ overlap when $\mathcal{I}_S \cap \mathcal{I}_T \neq \varnothing$ or $\mathcal{U}_S \cap \mathcal{U}_T \neq \varnothing$. Figure 1 shows the four scenarios deriving by the combination of user and items overlapping in cross-domain CF as identified by Cremonesi et al. in [2]:

- *No overlap.* There is no overlap between users and items, i.e., $\mathcal{U}_{ST} = \mathcal{U}_S \cap \mathcal{U}_T = \varnothing$ and $\mathcal{I}_{ST} = \mathcal{I}_S \cap \mathcal{I}_T = \varnothing$.
- *User overlap.* There are some shared users who have preferences for items in both domains, i.e., $\mathcal{U}_{ST} \neq \varnothing$.
- *Item overlap.* There are shared items rated by users from both the domains, i.e., $\mathcal{I}_{ST} \neq \varnothing$.
- *User and item overlap.* There is overlap between both users and items, i.e., $\mathcal{U}_{ST} \neq \varnothing$ and $\mathcal{I}_{ST} \neq \varnothing$.

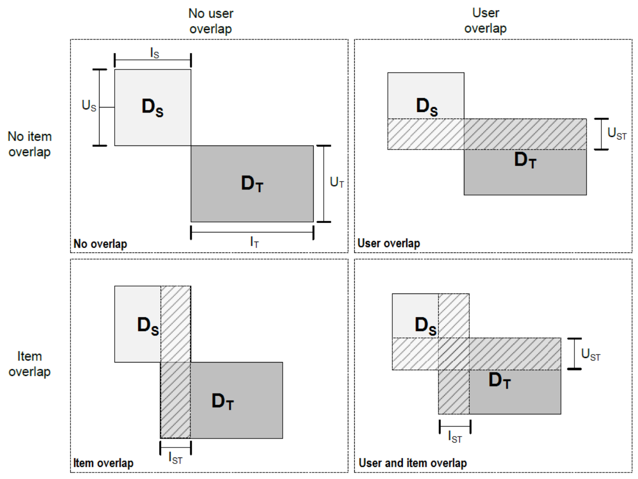

Figure 1: Scenarios of data overlap between user and item sets in two domains $\mathcal{D}_S$ and $\mathcal{D}_T$: *no overlap*, *user overlap*, *item overlap*, and *user and item overlap*.

2.1 The CBT algorithm

CBT is one of the recent CF methods based on rating pattern sharing which do not rely on any kind of overlap between domains [8, 7, 5]. The underlying hypothesis is that. even when users and items are different, related domains are likely to have user preferences sampled with the same population. Therefore, correlations may exist between preferences of groups of users for groups of items, which are referred to as the *rating patterns*. Rating patterns can act as a bridge that relates the involved domains, even though users and items may not overlap between the domains.

The CBT algorithm [7] extracts rating patterns from the source domain by simultaneously co-clustering users and items. Clustering is performed using a tri-factorization of the source rating matrix [3]. Then, knowledge is transferred through the *codebook*, a compact cluster-level matrix that is computed in the source domain taking the average rating of each user-item cluster. In the target domain, the sparse user-rating matrix is filled with ratings predicted using the shared codebook. The now filled user-rating matrix is then used to train a traditional CF algorithm.

The CBT algorithm consists of three steps: *codebook construction*, *target filling* and *CF training.*

In the **first step**, the source rating matrix R_S is factorized by using the orthogonal nonnegative matrix tri-factorization described in [3]

$$\min_{U_S \geq 0, V_S \geq 0, S \geq 0} \|R_S - U_S S V_S^\top\|_F^2 \qquad (1)$$
$$\text{s.t. } U_S^\top U_S = I, V_S^\top V_S = I$$

where $U_S \in \mathbb{R}_+^{n\times k}$,$V_S \in \mathbb{R}_+^{m\times l}$,$S \in \mathbb{R}_+^{k\times l}$, $\|\cdot\|_F^2$ is the Frobenius norm, n is the number of users, m the number of items, and k and l are the number of user and item clusters, respectively. Before the factorization, the zero elements of R_S are filled with the average user rating. After the factorization, matrices U_S and V_S are binarized to form two hard membership matrices. The codebook B is constructed as follows

$$B = [U_S^\top R_S V_S] \oslash [U_S^\top \mathbf{1}\mathbf{1}^\top V_S] \qquad (2)$$

where $\oslash$ is the entry-wise division and $\mathbf{1}$ is a vector of ones. The codebook contains the average rating of each co-cluster of users and items.

In the **second step**, the target rating matrix R_T is filled by expanding the values in the codebook. To this purpose, users and items in the target domains are first mapped to the co-clusters identified in the codebook by minimizing the following quadratic loss function

$$\min_{\substack{U_T \in \{0,1\}^{p\times k} \\ V_T \in \{0,1\}^{q\times l}}} \|R_T - U_T B V_T^\top\|^2 \quad (3)$$
$$\text{s.t } U_T\mathbf{1} = \mathbf{1}, V_T\mathbf{1} = \mathbf{1}$$

Each row in U_T or V_T is the cluster membership of the user or item. Note that, differently from what happens in the first step, the norm of this second minimization problem is computed only on the non-zero entries of the user-rating matrix. Once the user and item membership matrices U_T and V_T are obtained, all the zero elements of the target rating matrix R_T are filled with the appropriate elements of the codebook

$$R_T \leftarrow W \circ R_T + [\mathbf{1} - W] \circ \left[U_T B V_T^\top\right] \quad (4)$$

where $\circ$ denotes the entry-wise product and W is a binary mask of the same size as R_T used to mask zero elements.

In the **third step**, the filled target matrix can be used as the training data set for CF.

The authors performed experiments using a dense subset of the EachMovie dataset to compute the rating patterns, which were transferred to the more sparse MovieLens and BookCrossing datasets. They compared the accuracy of a traditional CF algorithm on the target domain with and without the transfer of the codebook from the source domain. The algorithm adopted for the comparison was a user-based k-nearest-neighbors (kNN) collaborative filtering based on Pearson correlation. According to the results, rating predictions were improved with the transfer of the codebook.

3. EVALUATION OF CBT

In this section we describe the methodology adopted to evaluate if the improvement of accuracy experimented by Li et al. in [7] with the CBT algorithm is due to a transfer of learning from source to target domain.

The CBT algorithm consists of three steps: (i) construction of the codebook from the source domain; (ii) filling of the missing ratings in the target domain by using the codebook; (iii) training a CF algorithm on the filled target domain. The goal of the second step is to reduce the sparsity on the target domain and to help the final CF algorithms in providing better recommendations.

We have modified the CBT algorithm by removing step (i) and by generating a codebook B that is not based on the source domain. In other words, we drop equations (1) and (2) and keep equations (3) and (4). We have tested three different methods to generate a "false" codebook:

- *Random* (RND): the codebook is created by randomly generating ratings uniformly distributed in the range $[3 \dots 5]$.
- *Global Effects* (GE): the codebook is created by estimating ratings with the global effects $r_{ui} = \mu + \mu_u + \mu_i$ described in [6], where μ is the overall average rating of the target user-rating matrix R_T and $\mu_u + \mu_i$ are deviations of user u and item i from the average, respectively.
- *Asymmetric SVD* (AsySVD): the codebook is created by estimating ratings with the matrix-factorization algorithm *Asymmetric SVD* described in [6] and applied to the target user-rating matrix R_T.

With both GE and AsySVD, the numbers of user and item clusters in the codebook B is equal to the size of the target user-rating matrix R_T (i.e., each cluster is composed by one user and one item).

3.1 Results

In this section we present the quality of the different approaches on **three standard datasets**: *MovieLens* [1], *Netflix* [1] and *BookCrossing* [9]. All the datasets are publicly available. Ratings in the MovieLens and Netflix datasets are in a 1-5 scale. Ratings in the Book-Crossing dataset are in a 1-10 scale and have been normalized to a 1-5 scale. All the three datasets have been used as target domains, while only Netflix and MovieLens have been used as source domains. We performed in total **four experiments**, as each of the two source datasets has been used in conjunction with the remaining two datasets.

The evaluation has been performed with a methodology similar to the one adopted in the CBT paper [7]. We randomly selected 500 users with at least 40 ratings and 1000 items from each dataset. Each dataset has been used as either the source or the target domain. For each target dataset, 200 users have been randomly removed to form the test set. For each user in the test set, all but 15 ratings are randomly removed from her/his profile and used as ground truth for the evaluation of mean absolute error (MAE) and root mean squared error (RMSE). The evaluation has been performed 10 times, each time sampling a different set of test users. Results are reported in Table 1 and are the average over the 10 repetitions. For lack of space, not all the combination of source/target domains are reported. Each column refers to a different algorithm.

- kNN: user-based kNN with Pearson similarity. Recommendations are performed on the target domain, without using the source domain. The size of the neighborhood is $k = 20$ users. In order to improve the accuracy of recommendations, missing values in the user-rating matrix are filled in with average user rating.
- CBT: the codebook transfer method described in [7]. The number of both user and item clusters is 50.
- RND: the "false" codebook is filled with random ratings. The number of clusters is the same as with CBT.
- AsySVD: the "false" codebook is filled with ratings generated with the Asymmetric SVD matrix-factorization algorithm. The number of user and item clusters is equal to the size of the target user-rating matrix.
- GE: the "false" codebook is filled with ratings generated with the global effects method. The number of user and item clusters is equal to the size of the target user-rating matrix.

source: MovieLens → target: BookCrossing					
	kNN	CBT	RND	AsySVD	GE
MAE	0.5216	**0.5064**	**0.4963**	**0.5089**	**0.5109**
RMSE	0.4736	0.4492	**0.4380**	0.4556	0.4578

source: MovieLens → target: Netflix					
	kNN	CBT	RND	AsySVD	GE
MAE	0.7285	**0.7090**	**0.7047**	**0.7085**	**0.7124**
RMSE	0.8630	**0.8208**	**0.8149**	**0.8245**	**0.8313**

source: Netflix → target: BookCrossing					
	kNN	CBT	RND	AsySVD	GE
MAE	0.5580	**0.5456**	**0.5388**	**0.5474**	**0.5486**
RMSE	0.5437	**0.5231**	**0.5113**	**0.5269**	**0.5290**

source: Netflix → target: MovieLens					
	kNN	CBT	RND	AsySVD	GE
MAE	0.7640	**0.7446**	**0.7425**	**0.7397**	**0.7417**
RMSE	0.9388	**0.8988**	**0.8932**	**0.8915**	**0.8935**

Table 1: Accuracy of different methods. Bold numbers are significantly different from the baseline method kNN.

As expected, CBT clearly outperforms the baseline method for all the datasets. Surprisingly, the experiments with the random codebook present an accuracy which is identical to the accuracy of the CBT method, even in the lack of knowledge transfer from source to target domain.

4. DISCUSSION AND CONCLUSIONS

In this section we try to investigate the surprisingly results obtained by the random codebook. By design, the random codebook does not transfer any knowledge from the source domain into the target domain. Therefore, the increase of accuracy introduced by the random algorithm cannot be explained in terms of transfer of knowledge and we need to look for a different explanation.

The insight of the solution of this problem is from the second step of the CBT, described in (3). The goal of this step is to map users and items in the target domain to the clusters derived in the source domain (in the case of a codebook correctly created in the source domain). The structure of the optimization problem (3) closely resemble regularized SVD matrix-factorization, as the minimization is performed only on the non-zero elements of the user-rating matrix [6]:

$$\min_{U_T \geq 0, V_T \geq 0} \|R_T - U_T V_T^\top\|^2$$

The main differences are the constraints on the U_T and V_T matrices and the presence of the codebook B.

We believe that the second step of the codebook transfer, which is supposed to simply map users and item to clusters, is in fact equivalent to the training of a matrix-factorization algorithm, provided that matrix B (the codebook) has a wide enough range of ratings to account for the constraints imposed on U_T and V_T.

We also believe that the enrichment of the target domain performed by (4) is equivalent to performing recommendations with the trained matrix-factorization model.

We finally believe that the last step of the CBT method (i.e., running the kNN algorithm over the enriched target matrix) does not provide any additional improvement to the quality of the estimations, as it is well know in the literature that matrix-factorization CF outperform user-based CF in terms of accuracy [6]. Our believes are confirmed by the last two columns of Table 1, showing how codebooks created by running CF algorithms on the target domain still provide a significant reduction of the errors, even if the codebooks do not transfer any knowledge from the source domain. Our findings prove that the modified CBT method is able to improve the accuracy of recommendations even when there is no transfer of knowledge from the source domain, but they do not directly disprove that the CBT method is able to transfer knowledge between non-overlapping domains.

Still, by applying Occam's razor, we should prefer the more general explanation and derive the conclusion that *CBT is not able to transfer knowledge between non-overlapping domains, as the experimented increased accuracy is originated by the matrix-factorization performed by equations* (3) *and* (4).

5. REFERENCES

[1] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[2] P. Cremonesi, A. Tripodi, and R. Turrin. Cross-domain recommender systems. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 496–503. IEEE, 2011.

[3] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[4] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 25–32. ACM, 2011.

[5] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo. Cross-domain recommendation via cluster-level latent factor model. In *Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2013.

[6] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[7] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI*, volume 9, pages 2052–2057, 2009.

[8] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 617–624. ACM, 2009.

[9] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32, New York, NY, USA, 2005. ACM.