Cold-Start Management with Cross-Domain Collaborative Filtering and Tags

Manuel Enrich, Matthias Braunhofer, and Francesco Ricci

Free University of Bozen - Bolzano,
Piazza Domenicani 3, 39100 Bolzano, Italy
{menrich,mbraunhofer,fricci}@unibz.it

Abstract. Recommender systems suffer from the new user problem, i.e., the difficulty to make accurate predictions for users that have rated only few items. Moreover, they usually compute recommendations for items just in one domain, such as movies, music, or books. In this paper we deal with such a cold-start situation exploiting cross-domain recommendation techniques, i.e., we suggest items to a user in one target domain by using ratings of other users in a, completely disjoint, auxiliary domain. We present three rating prediction models that make use of information about how users tag items in an auxiliary domain, and how these tags correlate with the ratings to improve the rating prediction task in a different target domain. We show that the proposed techniques can effectively deal with the considered cold-start situation, given that the tags used in the two domains overlap.

Keywords: Collaborative filtering, cross-domain recommendation, matrix factorization, tags.

1 Introduction

Recommender systems (RSs) are software tools that address the information overload problem by retrieving and suggesting items that are estimated as relevant for a user, based on her user profile. However, most of the available RSs [1] suffer from the data sparseness problem caused by the fact that users usually rate only a few items. This is especially true for users that have just joined the system and have not provided yet many ratings. To address this problem, researchers have considered cross-domain scenarios, i.e., have attempted to reuse users' knowledge in an auxiliary and better known domain in order to improve the accuracy of recommendations in another, less known, target domain [2]. The key challenge in cross-domain recommendation is to discover useful relationships among items or users in different domains, e.g., using similarities between items, or (as we will show in this paper) using the similarities of the conditions under which the items in the different domains are rated. Usually, the considered domains are heterogeneous (e.g., music vs. places of interest), making it difficult to find relationships or links between them.

In this research, we leverage user-assigned tags as a "bridge" between different domains. Tags have been shown in previous research to be useful for matching

items in one domain with those in others [3,4,5]. Our proposed technique relies on the usual overlap between the tag vocabularies used in different domains. For instance, the tag "romantic" could be used to describe a movie, a place of interest or a song. Hence, if for instance a RS targeted to a particular domain is able to learn, in an auxiliary domain, that a tag has a positive effect on the ratings, i.e., that when the tag is present the ratings are generally higher, then it could be possible to transfer this dependency from the auxiliary domain to the target one.

In this paper, we present three novel cross-domain rating prediction models, named as UserItemTags, UserItemRelTags and ItemRelTags, that are able to use tagging and rating data in an auxiliary domain to support rating prediction in a target domain for a completely new set of users. UserItemTags and UserItemRelTags predict a target user rating by considering the tags this user has assigned to the target item. While, ItemRelTags exploits the tags in a more general way, i.e., it considers all tags assigned by any user on the target item to compute rating predictions. Hence, this last algorithm does not use the knowledge of how the target user has tagged the target item to generate a rating prediction.

We have formulated the following hypothesis: the information about how users tag items in a particular domain can be exploited to improve the rating prediction accuracy in a completely different domain. To evaluate this hypothesis, we have carried out a series of tests using the MovieLens and the LibraryThing datasets, and have compared the results to those obtained by a state-of-the-art single domain recommendation algorithms based on matrix factorization [6].

The rest of the paper is structured as follows: in section 2 we position our work with respect to the state of the art. Section 3 presents our proposed cross-domain rating prediction models. Section 4 describes the experiments that we have performed in order to evaluate our models, and discusses the obtained results. Finally, conclusions and directions for future work of the presented approach are pointed out in section 5.

2   Related Work

As shown by [7], cross-domain recommendation techniques can tackle cold-start problems in collaborative filtering. Four methods for cross-domain collaborative filtering are there identified: centralized prediction, distributed peer identification, distributed neighborhood formation, and distributed prediction. However, differently from our work, they consider scenarios where the cross-domain recommenders do share some users, i.e., there are users that have rated items in several domains. Producing cross-domain recommendations for a new user in a target domain, without having any user ratings in auxiliary domains (as in our case) is more challenging since only item relationships across domains can be exploited.

Another example of a cross-domain recommender system developed to overcome cold-start problems is Tag-induced Cross-Domain Collaborative Filtering

(TagCDCF) [5]. TagCDCF exploits shared tags to link different domains through an extended matrix factorization framework. In TagCDCF, first the user-item matrices of different domains are factorized into domain-specific latent user and item features. Then, the latent features are linked across domains using tag-induced cross-domain similarities. The cross-domain scenario considered in [5] in similar to the one we have studied (there are not common users or items across domains), however, their approach requires the target user to have tagged several items in order to obtain accurate similarities between the target user in one domain and other users in the auxiliary domain.

Without relying on cross-domain techniques, one can tackle cold-start problems by exploiting, in addition to the ratings, other kinds of data relating users and items. For instance, SVD++ [6] extends the popular SVD matrix factorization model by exploiting implicit feedback. In SVD++ the factor-based profile of a user is additionally affected by the items that the user simply bought or browsed. This is achieved by introducing a second set of item-related factors' vectors that are learned on the base of the set of items that the users browsed or purchased.

A similar extension to SVD is presented in [8]; it exploits contextual factors (e.g., budget, time of the day, weather) in order to alter the rating prediction in a places of interest RS. Here the system learns to which extent the presence of a contextual condition influences the ratings, and this knowledge is used when a new recommendation is to be made. Our models generalize their idea: instead of using pre-defined contextual parameters, our proposed models can use any kind of user-generated textual information assigned to items to improve the recommendation.

In conclusion, the two aforementioned approaches [6,8] are similar to the one described in this paper: they both extend SVD considering additional item and user knowledge in order to deal with cold-start problems. However, to learn the prediction models, they require an extensive set of training data (i.e., browsing / purchase history, ratings in context) that is specific to the application's target domain, whereas our proposed models can improve the rating prediction task in a target domain by just re-using knowledge about tags usage acquired in the target domain as well as in a totally different domain, provided that there is an overlap between the set of tags used in the two domains.

3   Tag-Based Rating Prediction Models

This section describes the tag-based rating prediction models that we have developed to provide cross-domain recommendations. The underlying intuition is that tags could be used to improve the item model computed in matrix factorisation models. In fact, the tags assigned by users to items provide additional information about that item's rating. In our models we rely on tag applications for modelling the item's profile, i.e., how much a item loads the factors' model. Once the information about the impact of a tag is captured in one domain, we conjecture that it can be re-used to support rating prediction in a totally new

domain (e.g., composed by totally different items and users), as long as there
is an overlap of the tag vocabularies in the two domains. In other words, we
conjecture that the effect of tags on the factor model of items is cross-domains.

3.1 UserItemTags

The first model is based on the idea that the user ratings for an item may
be dependent on the specific tags the user attached to the item. This means
that the model, when generating a rating prediction, is considering the tags
used by the target user requesting the recommendation. To exploit this model
we assume therefore that the user tagged an item, without providing a rating,
and we exploit these tags to better predict her rating. This happens in many
situations: a common example is the Delicious[1] social bookmarking website, in
which users can apply tags to their bookmarks, but are not asked to rate the
bookmarked website.

Given a user u, an item i and the set of tags $T_u(i)$ assigned by u to i, UserItem-
Tags predicts a rating using the following rule:

$$\hat{r}_{ui} = p_u \cdot (q_i + \frac{1}{|T_u(i)|} \sum_{t \in T_u(i)} y_t), \tag{1}$$

where p_u, q_i and y_t are the latent factor vectors associated with the user u, the
item i and the tag t, respectively. The model parameters are learned, as it is com-
mon in matrix factorization [6], by minimising the associated regularised squared
error function through stochastic gradient descent. This is done by looping over
all known ratings in K, computing:

- $p_u \leftarrow p_u + \gamma \cdot [(q_i + \frac{1}{|T_u(i)|} \cdot \sum_{t \in T_u(i)} y_t) \cdot e_{ui} - \lambda \cdot p_u]$
- $q_i \leftarrow q_i + \gamma \cdot (p_u \cdot e_{ui} - \lambda \cdot q_i)$
- $\forall t \in T_u(i) : y_t \leftarrow y_t + \gamma \cdot (p_u \cdot \frac{1}{|T_u(i)|} \cdot e_{ui} - \lambda \cdot y_t)$

3.2 UserItemRelTags

UserItemRelTags is a variant of the previous model: UserItemTags. Its definition
is based on the intuition that tags have different relevances when performing rat-
ing prediction. For example, a tag assigned to a movie could be the name of the
main actor(s), the year of production or a textual label that is only meaningful
to the user applying that tag (e.g., the occasion when the user watched that
movie). This introduces a huge variety of tags, and only part of them can be
useful (i.e., relevant) when predicting users' ratings. UserItemRelTags considers
only a set of relevant tags, minimising the "noise" introduced by irrelevant tags.
To assess whether a tag is relevant or not, we used the Wilcoxon rank-sum test
(95% confidence) and compared, for each tag, the distribution of the ratings
assigned with or without the presence of the tag. Hence, a tag is judged statis-
tically relevant if the average of all the users' ratings where the tag is present is

---

[1] Delicious: `https://delicious.com/`

significantly different from the average of all the users' ratings where the tag is not used. We chose the Wilcoxon rank-sum test over the more commonly used two-samples t-test because we can not make any assumption about the normality of the distributions of the ratings. We observe that more sophisticated techniques for filtering out irrelevant tags can be implemented; for instance, as shown in [9], one can categorize tags in different semantic groups and then treat them as having different relevance weights.

Denoting with $TR_u(i)$ the set of relevant tags assigned by user u on item i, UserItemRelTags predicts user ratings as follows:

$$\hat{r}_{ui} = p_u \cdot (q_i + \frac{1}{|TR_u(i)|} \sum_{t \in TR_u(i)} y_t). \tag{2}$$

Model parameters are determined in the same way as for UserItemTags.

3.3   ItemRelTags

The last model, ItemRelTags, does not rely on the tags assigned by the target user to the target item but it considers all the tags applied to the target item by any user. This allows to overcome the main limitation of the two previous models, which is the inability to provide a rating prediction for an item that was not tagged by the target user. In fact, ItemRelTags requires only the knowledge of $TR(i)$, which is the set of relevant tags applied to item i by any user. The relevance of a tag is estimated using a Wilcoxon rank-sum test as in UserItem-RelTags. The intuition motivating this model is that the ratings of an item are affected by all the relevant tags applied to it. Moreover, we assume that tags have a common influence for any user (i.e., they are used in a non-personalized way). This assumption is clearly imprecise, but in the absence of the information about how the target user has tagged the target item, it is all one can do. We will test experimentally whether there is a dependence of a user rating on the full set of the tags given to the target item by the users population.

Since this model uses all the relevant tags assigned by any user to the target item, the same tag can appear multiple times. We may imagine that if a tag has been assigned several times to a single item, then it better characterizes the item. Therefore, we have also exploited the tag usage frequency into the model. Let us call TRo_i the relevant tag occurrences for item i, i.e., all the relevant tags applied to the item, duplicates included, and $TRo_i(t)$ representing the relevant tag occurrences of tag t in item i. Then, the prediction rule for ItemRelTags is as follows:

$$\hat{r}_{ui} = p_u \cdot (q_i + \frac{1}{|TRo_i|} \sum_{t \in TR(i)} TRo_i(t)y_t). \tag{3}$$

Even in this case, the model parameters are learned by stochastic gradient descent optimization of the associated squared error function.

4   Experimental Evaluation

In order to evaluate the proposed rating prediction models, we performed two experiments. The goal of the first experiment was to measure the rating prediction accuracy of the models in a cross-domain scenario. In the second experiment, we assessed the quality of rating prediction using only rating and tagging data in a single domain and compared it with the previous results (exploiting cross-domain data).

4.1   Cross-Domain Recommendations

In this section, we present the results of the evaluation study of our models in the cross-domain scenario. We first describe the datasets used in the evaluation, then the experimental design, and finally we present the performance of our rating prediction models compared with SVD.

Datasets. We have evaluated the proposed models using two freely available datasets: MovieLens[2], containing 10 million ratings, and LibraryThing[3], containing over 700 thousand ratings. In both datasets, the ratings are expressed on a scale from 1 to 5, with steps of 0.5. Moreover, MovieLens contains 100,000 tag assignments applied by 72,000 users on 10,000 movies, and LibraryThing contains 2 million tag assignments applied by over 7,000 users on 37,000 books.

Many ratings contained in MovieLens do not contain tag assignments, i.e., the user only rated the item. When computing a prediction without exploiting any tagging information our models behaves exactly as SVD. Since in the tests we wanted to investigate the benefit produced by the tagging data in cross-domain predictions, we therefore considered only the ratings in which at least one tag was used. In this way we obtained a total of 24,565 ratings. In order to limit the effect produced by the variation in the quantity of the ratings we considered in the LibraryThing domain only its first 24,564 ratings, exactly the same number of ratings with tags found in MovieLens. In this subset of the available ratings, the tags in MovieLens covered 29.31% of the tags used in LibraryThing, and the tags from LibraryThing covered 14.54% of the tags used in MovieLens. In LibraryThing there are less distinct tags and they are also used more than in MovieLens. We conjecture that these differences are important to estimate how useful an auxiliary domain can be in the prediction of the ratings in the target domain. More details about the datasets are provided in table 1.

Evaluation Design. For each tested model we have obtained two results: one using MovieLens as target and LibraryThing as auxiliary domain, and another using LibraryThing as target and MovieLens as auxiliary domain. In order to ensure the reliability of our results, we cross validated them. For the cross validation process, we shuffled the target domain rating data and then split it in

---

[2] MovieLens dataset: `http://www.grouplens.org/node/73`
[3] LibraryThing dataset: `http://www.librarything.com/`

Table 1. Features of the MovieLens and LibraryThing datasets used to test the models

	MovieLens	LibraryThing
Total number of ratings	24,564	24,564
Unique users	2,026	283
Unique items	5,088	12,554
Unique tags	9,486	4,708
Tag assignments	44,805	78,239
Average ratings per user	12.12	86.80
Average tags per rating	1.82	3.18
% of tags overlapping with LibraryThing / MovieLens	14.54	29.31

ten parts (ten-folds cross validation). In each validation iteration, we used one of the obtained splits as test and the remaining data as training set. In order to test the behaviour of the developed models with different amounts of data in the target domain (representing therefore a recommender system in which users and ratings are incrementally introduced), we split the training candidates data again into ten parts. Each of the obtained parts contains therefore a set of non overlapping ratings (each containing 10% of the total training data), that we have used incrementally in our tests. Hence, for estimating the system performance with small knowledge of the target domain we have used 10% of the training data, i.e., 24,564 * 9/100 = 2210 ratings. For simulating a situation where some more knowledge of the target domain is available, we used 20% of the training data, i.e., the first two parts of the training set, etc. The selected amount of target domain data is then extended with the full set of the auxiliary data to obtain the actual training data of the predictive models. This process was repeated ten times, allowing to test our models with each of the original ten target splits as testing data.

We have compared our models with SVD. SVD cannot exploit additional information coming from the auxiliary domain because the set of users are disjoint. This is peculiar to the cross-domain situation that we consider in this paper. Hence, SVD is only able to use the training data in the target domain.

The model parameters (i.e., dimensionality f, learning rate γ and regularization λ) that yielded the best prediction results were obtained by the Nelder-Mead simplex algorithm [10] using the union of the MovieLens and LibraryThing datasets, and were as shown in Table 2.

Table 2. Model parameters obtained using the Nelder-Mead algorithm (cross-domain)

	γ	λ	f
SVD	0.037	0.012	18
UserItemTags	0.018	0.013	10
UserItemRelTags	0.015	0.02	10
ItemRelTags	0.031	0.022	10

Evaluation Results. The obtained average mean absolute errors (MAEs) show that the proposed models, which are based on tagging information from an auxiliary domain, outperform SVD in terms of prediction accuracy (see Figure 1 and Figure 2). In these figures, we marked with a black (grey) circle when we obtained a significant better (worse) result than the baseline SVD (using a t-test at 95%), and no circle when there is no statistical difference. As can be seen, in most of the cases the usage of rating and tagging data in the auxiliary domain reduces the prediction error in the target domain. However, when the target is MovieLens the tagging data from LibraryThing was not able to improve the prediction accuracy in the very cold start situation, i.e., when only 10% or 20% of the ratings in MovieLens were user. We believe that this result is explained by the fact that the tags from the auxiliary domain (i.e., LibraryThing) only cover a small part of the tags used in the target domain (i.e., MovieLens). This makes the knowledge transfer more difficult, since only a small part of the dependency between ratings and tags learned in the auxiliary domain can be successfully exploited. Moreover, when only 10% of the rating in the target domain are used, the predictive model is much more influenced by the relationships between tags and ratings that are present in the auxiliary domain.
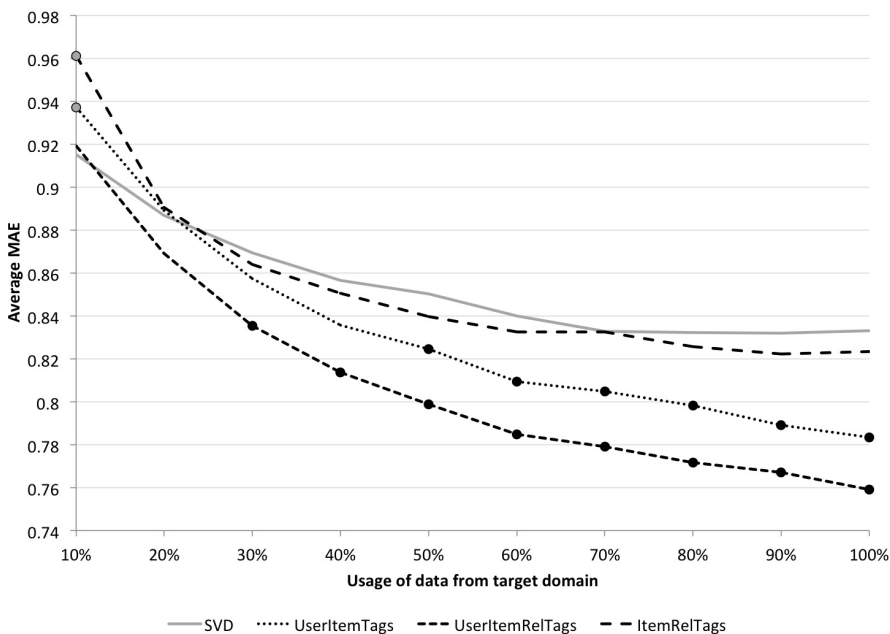

Fig. 1. Average MAEs using MovieLens as target domain and LibraryThing as auxiliary domain
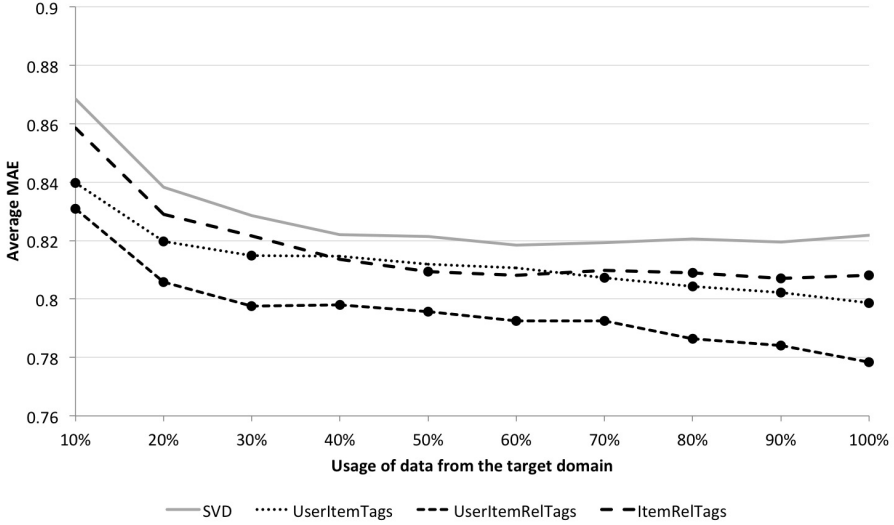
Fig. 2. Average MAEs using LibraryThing as target domain and MovieLens as auxiliary domain

4.2 Single-Domain Recommendations

This section describes the experimental design and the results of the evaluation of the proposed tag-based rating prediction models in a single-domain scenario. We aimed at verifying whether the proposed models can provide good predictions using only rating and tagging data in the target domain. Moreover, since we have observed that UserItemRelTags always performs better than ItemUserTags we do not show here the results for UserItemTags.

Evaluation Design. In order to ensure the reliability of our results, we cross validated them as we did in the cross-domain experiment: we shuffled the data in the target domain and then split it in ten parts (ten-folds validation). In each validation iteration, we used one of the obtained split as test, and the remaining data as training set. We split the training data into ten further parts, to obtain a set of non overlapping incremental data segments to be used as training. This process was repeated ten times, allowing to test our models with each of the original ten target splits as testing data.

We used again SVD as baseline model. Moreover, we have compared the accuracies of the prediction models in the single domain with those previously obtained in the cross-domain scenario. The goal was to understand whether the improvement with respect to SVD is due to the knowledge transferred across domains, or by the additional tagging information in the target domain.

Table 3. Model parameters obtained using the Nelder-Mead algorithm (single domain)

	MovieLens			LibraryThing		
	γ	λ	f	γ	λ	f
SVD	0.014	0.016	17	0.016	0.016	16
UserItemTags	0.02	0.015	10	0.02	0.01	15
UserItemRelTags	0.023	0.018	10	0.02	0.01	15
ItemRelTags	0.015	0.02	10	0.02	0.02	16
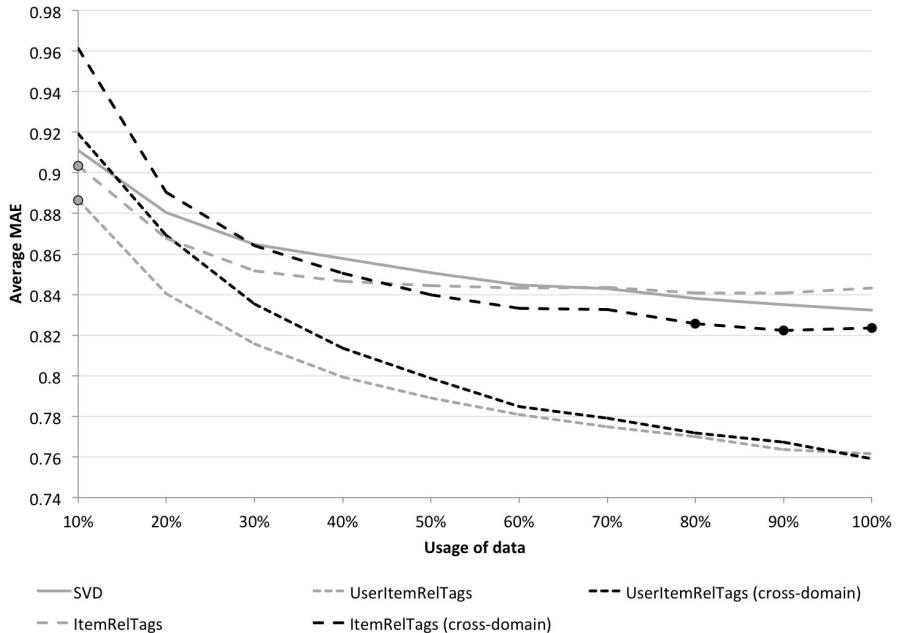
Fig. 3. Comparison of models' MAEs - single vs. cross domain (MovieLens target)

Like in the cross-domain case, also here the model parameters have been obtained using the Nelder-Mead approach [10] but separately for each data set (see Table 3).

Evaluation Results. The obtained results are shown in Figure 3 and 4. In these figures, a black (grey) circle is used to indicate that the results obtained in the cross domain situation are significantly better (worse) than the ones obtained by the same model in the single domain situation. For better visibility, as mentioned above, we omit the curves of UserItemTags which always performed worse than UserItemRelTags. It can be noted that the tagging information always yields a benefit compared with SVD. Comparing single domain vs. cross domain tagging usage the situation is again different in the two data sets. When the target is MovieLens the single domain approach is normally better.

While in LibraryThing the auxiliary domain tagging data are useful, especially in the cold start situation, i.e., when a small quantity of training data from the target domain is provided. It is quite surprising to note that in MovieLens Item-RelTags is better in the cross-domain application than in the single domain one only when the largest portion of the ratings in MovieLens are used. A possible explanation to this is given by the fact that the auxiliary domain only covers part of the tags used in the target domain, and therefore when only a small amount of data from the target domain is used in the training phase the models are not able to successfully exploit the knowledge acquired from the auxiliary domain. However, this is only a conjecture that deserves a more extensive evaluation.
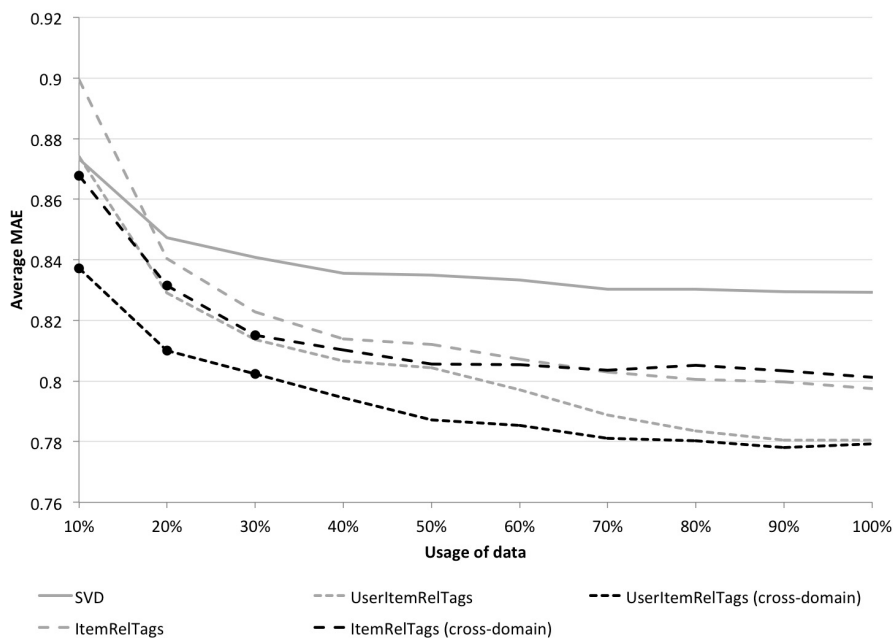

Fig. 4. Comparison of models' MAEs - single vs. cross domain (LibraryThing target)

5   Conclusions and Future Work

In this paper we have presented a set of novel cross-domain recommender system models, called UserItemTags, UserItemRelTags and ItemRelTags. They are shown to be able to improve the accuracy of the rating prediction on a target domain using rating and a tagging data coming from an auxiliary domain, even if the users in the two domains are disjoint. In these models, the knowledge transfer across domains is performed using information about which items have been annotated with certain tags. We have formulated the following experimental hypothesis: the information about how users tag items in a domain can be

exploited in order to improve the rating prediction accuracy in a totally different domain. Results obtained from a series of tests conducted on the MovieLens and LibraryThing datasets confirmed this hypothesis.

The proposed cross-domain recommendation techniques are new, and there is a number of research questions left unaddressed. First of all, we should better correlate algorithm performance to the characteristics of the data sets (sparsity, distribution of tags, overlap of tags between domains). Secondly, the performance of our proposed models on other datasets should be assessed and a comparison with other cross-domain recommenders is in order [11,12,5]. Moreover, we are interested in better understanding the conditions when the tag-based models can be exploited, e.g., in context-aware recommender systems, and if these techniques could be used to generate more diverse recommendations.

References

1. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer (2011)
2. Cremonesi, P., Tripodi, A., Turrin, R.: Cross-domain recommender systems. In: 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW), Vancouver, BC, Canada, December 11, pp. 496–503 (2011)
3. Kaminskas, M., Ricci, F.: Location-adapted music recommendation using tags. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 183–194. Springer, Heidelberg (2011)
4. Braunhofer, M., Kaminskas, M., Ricci, F.: Recommending music for places of interest in a mobile travel guide. In: Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 253–256. ACM (2011)
5. Shi, Y., Larson, M., Hanjalic, A.: Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 305–316. Springer, Heidelberg (2011)
6. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 145–186. Springer (2011)
7. Berkovsky, S., Kuflik, T., Ricci, F.: Distributed collaborative filtering with domain specialization. In: Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 33–40. ACM (2007)
8. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing 16(5), 507–526 (2012)
9. Cantador, I., Konstas, I., Jose, J.M.: Categorising social tags to improve folksonomy-based recommendations. Web Semantics: Science, Services and Agents on the World Wide Web 9(1), 1–15 (2011)
10. Nelder, J.A., Mead, R.: A simplex method for function minimization. The Computer Journal 7(4), 308–313 (1965)
11. Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. User Modeling and User-Adapted Interaction 23(2-3), 169–209 (2013)
12. Wang, W., Chen, Z., Liu, J., Qi, Q., Zhao, Z.: 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, pp. 10–17. ACM (2012)