Exploiting Social Tags in Matrix Factorization Models for Cross-domain Collaborative Filtering

Ignacio Fernández-Tobías, Iván Cantador
Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049 Madrid, Spain
{ignacio.fernandezt, ivan.cantador}@uam.es

ABSTRACT

Cross-domain recommender systems aim to generate or enhance personalized recommendations in a target domain by exploiting knowledge (mainly user preferences) from other source domains. Due to the heterogeneity of item characteristics across domains, content-based recommendation methods are difficult to apply, and collaborative filtering has become the most popular approach to cross-domain recommendation. Nonetheless, recent work has shown that the accuracy of cross-domain collaborative filtering based on matrix factorization can be improved by means of content information; in particular, social tags shared between domains. In this paper, we review state of the art approaches in this direction, and present an alternative recommendation model based on a novel extension of the SVD++ algorithm. Our approach introduces a new set of latent variables, and enriches both user and item profiles with independent sets of tag factors, better capturing the effects of tags on ratings. Evaluating the proposed model in the movies and books domains, we show that it can generate more accurate recommendations than existing approaches, even in cold-start situations.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering*. G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra – *singular value decomposition*.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Recommender systems, collaborative filtering, cross-domain recommendation, social tagging.

1. INTRODUCTION

Recommender systems [2] have been successfully used in numerous domains and applications to identify potentially relevant items for users according to their preferences (tastes, interests and goals). Examples include suggested movies and TV programs in Netflix[1], music albums in Last.fm[2], and books in Barnes&Noble[3].

[1] Netflix online movies & TV shows provider, http://www.netflix.com
[2] Last.fm music discovery service, http://www.lastfm.com
[3] Barnes&Noble retail bookseller, http://www.barnesandnoble.com

CBRecSys 2014, October 6, 2014, Silicon Valley, CA, USA.

Even though the majority of recommender systems focus on a single domain or type of item, there are cases in which providing the user with *cross-domain recommendations* could be beneficial. For instance, large e-commerce sites like Amazon[4] and eBay[5] collect user feedback for items from multiple domains, and in social networks users often share their tastes and interests on a variety of topics. In these cases, rather than exploiting user preference data from each domain independently, recommender systems could exploit more exhaustive, multi-domain user models that allow generating item recommendations spanning several domains. Furthermore, exploiting additional knowledge from related, auxiliary domains could help improve the quality of item recommendations in a target domain, e.g. addressing the cold-start and sparsity problems [7].

These benefits rely on the assumption that there are similarities or relations between user preferences and/or item attributes from different domains. When such correspondences exist, one way to exploit them is by *aggregating knowledge* from the involved domain data sources, for example by merging user preferences into a unified model [1], and by combining single-domain recommendations [3]. An alternative way consists of *transferring knowledge* from a source domain to a target domain, for example by sharing implicit latent features that relate source and target domains [15][17], and by exploiting implicit rating patterns from source domains in the target domain [9][14].

In either of the above cases, most of the existing approaches to cross-domain recommendation are based on collaborative filtering, since it merely needs rating data, and does not require information about the users' and items' characteristics, which are usually highly heterogeneous among domains.

However, inter-domain links established through content-based features and relations may have several advantages, such as a better interpretability of the cross-domain user models and recommendations, and the establishment of more reliable methods to support the knowledge transfer between domains. In particular, social tags assigned to different types of items –such as movies, music albums, and books–, may act as a common vocabulary between domains [6][17]. Hence, as domain independent content-based features, tags can be used to overcome the information heterogeneity across domains, and are suitable for building the above mentioned inter-domain links.

In this paper, we review state of the art cross-domain recommendation approaches that utilize social tags to exploit knowledge from an auxiliary source domain for enhancing collaborative filtering rating predictions in a target domain.

[4] Amazon e-commerce website, http://www.amazon.com
[5] eBay consumer-to-consumer website, http://www.ebay.com

Specifically, we focus on several extensions of the matrix factorization technique proposed in [6], which incorporates latent factors related to the users' social tags. By jointly learning tag factors in both the source and target domains, hidden correlations between ratings and tags in the source domain can be used in the target domain. Hence, for instance, a movie recommender system may estimate a higher rating for a particular movie tagged as *interesting* or *amazing* if these tags are usually assigned to books positively rated. Also, books tagged as *romantic* or *suspenseful* may be recommended to a user if it is found that such tags correlate with high movie ratings.

Enrich et al. [6] presented several recommendation models that exploit different sets of social tags when computing rating predictions, namely tags assigned by the active user to the item for which the rating is estimated, and all the tags assigned by the community to the target item. Despite their good performance, these models do have difficulties in cold-start situations where no tagging information is available for the target user/item.

In this paper, we propose a method that expands the users' and items' profiles to overcome these limitations. More specifically, we propose to incorporate additional parameters to the above models, separating user and item latent tag factors in order to capture the contributions of each to the ratings more accurately. Furthermore, by modeling user and item tags independently we are able to compute rating predictions even when a user has not assigned any tag to an item, or for items that have not been tagged yet. For such purpose, we adapt the gSVD++ algorithm [10] – designed to integrate content metadata into the matrix factorization process– for modeling social tags in the cross-domain recommendation scenario.

Through a series of experiments in the movies and books domains, we show that the proposed approach outperforms the state of the art methods, and validate the main contribution of this work: A model that separately captures user and item tagging information, and effectively transfers auxiliary knowledge to the target domain in order to provide cross-domain recommendations.

The reminder of the paper is structured as follows. In section 2 we review state of the art approaches to the cross-domain recommendation problem, focusing on algorithms based on matrix factorization, and on algorithms that make use of social tags to relate the domains of interest. In section 3 we provide a brief overview of matrix factorization methods for single-domain recommendation, and in section 4 we describe their extensions for the cross-domain recommendation case. In section 5 we present and discuss the conducted experimental work and obtained results. Finally, in section 6 we summarize some conclusions and future research lines.

2. RELATED WORK

Cross-domain recommender systems aim to generate or enhance personalized recommendations in a target domain by exploiting knowledge (mainly user preferences) from other source domains [7][19]. This problem has been addressed from various perspectives in several research areas. It has been faced by means of user preference aggregation and mediation strategies for the cross-system personalization problem in user modeling [1][3][16], as a potential solution to mitigate the cold-start and sparsity problems in recommender systems [5][17][18], and as a practical application of knowledge transfer techniques in machine learning [9][14][15].

We can distinguish between two main types of cross-domain approaches: Those that *aggregate* knowledge from various source domains to perform recommendations in a target domain, and those that *link* or *transfer* knowledge between domains to support recommendations in the target domain.

The knowledge aggregation methods merge user preferences (e.g. ratings, social tags, and semantic concepts) [1], mediate user modeling data exploited by various recommender systems (e.g. user similarities and user neighborhoods) [3][16], and combine single-domain recommendations (e.g. rating estimations and rating probability distributions) [3]. The knowledge linkage and transfer methods relate domains by common information (e.g. item attributes, association rules, semantic networks, and inter-domain correlations) [5][18], share implicit latent features that relate source and target domains [15][17], and exploit explicit or implicit rating patterns from source domains in the target domain [9][14].

Cross-domain recommendation models based on latent factors are a popular choice among knowledge linkage and transfer methods, since they allow automatically discovering and exploiting implicit domain relations within the data from different domains. For instance, Zhang et al. [20] proposed an adaptation of the matrix factorization model to include a probability distribution that captures inter-domain correlations, and Cao et al. [4] presented a method that learns similarities between item latent factors in different domains as parameters in a Bayesian framework. Aiming to exploit heterogeneous forms of user feedback, Pan et al. [15] proposed an adaptive model in which the latent features learned in the source domain are transferred to the target domain in order to regularize the matrix factorization there. Instead of the more common two-way decomposition of the rating matrix, Li et al. [14] used a nonnegative matrix tri-factorization to extract rating patterns –the so-called *codebook*– in the source domain. Then, rather than transferring user and item latent factors, the rating patterns are shared in the target domain and used to predict the missing ratings.

Despite the ability of matrix factorization models to discover latent implicit relations, there are some methods that use tags as explicit information to bridge the domains. Shi et al. [17] argued that explicit relations established through common social tags are more effective for such purpose, and used them to compute user-user and item-item cross-domain similarities. In this case, rating matrices from the source and target domains are jointly factorized, but user and item latent factors are restricted so that they are consistent with the tag-based similarities.

Instead of focusing on sharing user or item latent factors, Enrich et al. [6] studied the influence of social tags on rating prediction. More specifically, the authors presented a number of models based on the well-known SVD++ algorithm [11], to incorporate the effect of tag assignments into rating estimations. The underlying hypothesis is that information about how users annotate items in the source domain can be exploited to improve rating prediction in a different target domain, as long as a set of common tags between the domains exists. In all the proposed models, tag factors are added into the latent item vectors, and are then combined with user latent features to compute rating estimations. The difference between these models is the set of tags considered for rating prediction. Two of the proposed models use the tags assigned by the user to a target item, and the other model takes the tags of the whole community into account. We note that the first two models require the active user to tag, but not rate the item in the target domain. In all the models, the transfer of knowledge is performed through the shared tag factors in a collective way, since these factors are learned jointly for the source and the target domains. The results reported in the movies and books domains confirmed that shared knowledge can be effectively exploited to outperform single-domain rating predictions.

The model we propose in this paper follows the same line as Enrich et al. [6], in the sense that tags are directly integrated as latent factors into the rating prediction process, as opposed to Shi's and colleagues' approach [17], which estimates the ratings using only user and item factors. The main difference of our model with the approaches presented in [6] is the way in which the rating matrix is factorized. Rather than using a single set of tag factors to extend the item's factorization component, we introduce additional latent variables in the user component to separately capture the effect of tags utilized by the user and the tags assigned to the item. For this purpose, we adapt the gSVD++ algorithm [10], which extends SVD++ by introducing a set of latent factors to take item metadata into account for rating prediction. In this model, both user and item factors are respectively enhanced with implicit feedback and content information, which allows improving the accuracy of rating predictions.

3. OVERVIEW OF MATRIX FACTORIZATION METHODS

Since the proposed cross-domain recommendation model is built upon a matrix factorization collaborative filtering method, in this section we provide a brief overview of the well-known standard rating matrix factorization technique, and the SVD++ and gSVD++ algorithms, which extend the former by incorporating implicit user feedback and item metadata, respectively.

3.1 MF: Standard rating matrix factorization

Matrix factorization (MF) methods [8][12] are a popular approach to latent factor models in collaborative filtering. In these methods, the rating matrix is decomposed as the product of low-rank matrices of user and item latent features. In its most basic form, a factor vector $p_u \in \mathbb{R}^k$ is assigned to each user u, and a factor vector $q_i \in \mathbb{R}^k$ to each item i, so that ratings are estimated as:

$$\hat{r}_{ui} = q_i^T p_u + b_{ui} \tag{1}$$

where the term b_{ui} is a baseline estimate that captures the deviation of user and item ratings from the average, and is defined as:

$$b_{ui} = \mu + b_u + b_i \tag{2}$$

The parameter μ corresponds to the global average rating in the training set, and b_u and b_i are respectively the deviations in the ratings of user u and item i from the average. The baseline estimates can be explicitly defined or learned from the data. In the latter case, the parameters of the model are found by solving the following regularized least squares problem:

$$\begin{gathered}\min_{b_*, p_*, q_*} \sum_{(u,i)\in\mathcal{R}} \left(r_{ui} - \mu - b_u - b_i - q_i^T p_u\right)^2 \\ + \lambda\left(b_u^2 + b_i^2 + \|p_u\|^2 + \|q_i\|^2\right)\end{gathered} \tag{3}$$

In this formula, the parameter λ controls the amount of regularization to prevent high model variance and overfitting. The minimization can be performed by using gradient descent over the set $\mathcal{R}$ of observed ratings [8]. This method is popularly called SVD, but it is worth noticing that it is not completely equivalent to the *singular value decomposition* technique, since the rating matrix is usually very sparse and most of its entries are actually not observed.

For simplicity purposes, in the following we omit the baseline estimates. They, nonetheless, can be easily considered by adding the b_{ui} term into the rating estimation formulas.

3.2 SVD++: Adding implicit user feedback to the rating matrix factorization method

The main motivation behind the SVD++ algorithm, proposed by Koren [11][13], is to exploit implicit additional user feedback for rating prediction, since it is arguably to use a more available and abundant source of user preferences.

In this model, user preferences are represented as a combination of explicit and implicit feedback, searching for a better understanding of the user by looking at what items she rated, purchased or watched. For this purpose, additional latent factors are combined with the user's factors as follows:

$$\hat{r}_{ui} = q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j \right) \tag{4}$$

In the previous formula, $p_u \in \mathbb{R}^k, q_i \in \mathbb{R}^k, y_j \in \mathbb{R}^k$ represent user, item, and implicit feedback factors, respectively. $N(u)$ is the set of items for which the user u provided implicit preference, and k is the number of latent features.

Similarly to the SVD algorithm, the parameters of the model can be estimated by minimizing the regularized squared error loss over the observed training data:

$$\begin{gathered}\min_{p_*, q_*, y_*} \sum_{(u,i)\in\mathcal{R}} \left[r_{ui} - q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j \right) \right]^2 \\ + \lambda \left(\|p_u\|^2 + \|q_i\|^2 + \sum_{j\in N(u)} \|y_j\|^2 \right)\end{gathered} \tag{5}$$

Again, the minimization problem can be efficiently solved using stochastic gradient descent.

3.3 gSVD++: Adding item metadata to the rating matrix factorization method

The gSVD++ algorithm [10] further extends SVD++ considering information about the items' attributes in addition to the users' implicit feedback.

The model introduces a new set of latent variables $x_g \in \mathbb{R}^k$ for metadata that complement the item factors. This idea combined with the SVD++ algorithm leads to the following formula for computing rating predictions:

$$\hat{r}_{ui} = \left(q_i + |G(i)|^{-\beta} \sum_{g\in G(i)} x_g \right)^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j \right) \tag{6}$$

The set $G(i)$ contains the attributes related to item i, e.g. *comedy* and *romance* in the case of movie genres. The parameter β is set to 1 when the set $G(i) \neq \emptyset$, and 0 otherwise. We note that in the previous formula, both user and item factors are enriched with new uncoupled latent variables that separately capture information about the users and items, leading to a symmetric model with four types of parameters. Again, parameter learning can be performed by minimizing the associated squared error function with gradient descent:

$$\begin{gathered}\min_{p_*, q_*, x_*, y_*} \sum_{(u,i)\in\mathcal{R}} \left[r_{ui} - \left(q_i + |G(i)|^{-\beta} \sum_{g\in G(i)} x_g \right)^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j \right) \right]^2 \\ + \lambda \left(\|p_u\|^2 + \|q_i\|^2 + \sum_{g\in G(i)} \|x_g\|^2 + \sum_{j\in N(u)} \|y_j\|^2 \right)\end{gathered} \tag{7}$$

The use of additional latent factors for item metadata is reported to improve prediction accuracy over SVD++ in [10]. In this paper, we adapt this model to separately learn user and item tag factors, aiming to support the transfer of knowledge between domains.

4. TAG-BASED MODELS FOR CROSS-DOMAIN COLLABORATIVE FILTERING

In this section, we first describe the tag-based cross-domain collaborative filtering models presented in [6], which are an adaptation of the SVD++ algorithm, and next introduce our proposed model, which is built upon the gSVD++ algorithm.

4.1 Adaptation of SVD++ for Tag-based Cross-domain Collaborative Filtering

The main hypothesis behind the models proposed in [6] is that the effect of social tags on ratings can be shared between domains to improve the rating predictions in the target domain. In that work, three different adaptations of the SVD++ algorithm were explored that utilize tags as implicit user feedback to enhance the item factors, as opposed to user factors like in the original model.

The first of the algorithms proposed by Enrich et al. is the **UserItemTags** model, which only exploits the tags $T_u(i)$ that the active user u assigned to the target item i:

$$\hat{r}_{ui} = p_u^T \left(q_i + \frac{1}{|T_u(i)|} \sum_{t \in T_u(i)} y_t \right) \tag{8}$$

We note here that if the user has not tagged the item, i.e., $T_u(i) = \emptyset$, then the model corresponds to the standard matrix factorization technique. Also, even though the tag factors y_t are only combined with the item factors q_i, the user and item factorization components are not completely uncoupled, since the set $T_u(i)$ still depends on the user u.

An improvement over the model was also presented in [6], based on the observation that not all the tags are equally relevant (i.e. discriminative) to predict the ratings. The proposed alternative is to filter the tags in the set $T_u(i)$ that are not relevant according to certain criterion. In that work, the Wilcoxon rank-sum test is performed for each tag to decide if the mean rating significantly changes in the presence/absence of the tag in the dataset. In this model, rating predictions are computed in an analogous manner:

$$\hat{r}_{ui} = p_u^T \left(q_i + \frac{1}{|TR_u(i)|} \sum_{t \in TR_u(i)} y_t \right) \tag{9}$$

Here, the set $TR_u(i) \subseteq T_u(i)$ only contains those tags for which the p-value of the abovementioned test is $p < 0.05$. This method was called as **UserItemRelTags**.

As noted by the authors, the previous methods are useful when the user has tagged but not rated an item. However, these methods do not greatly improve over the standard matrix factorization technique in the cold-start situations where new users or items are considered. Aiming to address this limitation, a last approach was proposed, the **ItemRelTags** model:

$$\hat{r}_{ui} = p_u^T \left(q_i + \frac{1}{|TR(i)|} \sum_{t \in TR(i)} n_t y_t \right) \tag{10}$$

Now, the set $TR(i)$ contains all the relevant tags assigned by the whole community to the item i, with possible repetitions. Tags that appear more often contribute with more factors to the prediction, and n_t is the number of times tag t was applied to item i. In this case, the normalization factor is $|TR(i)| = \sum_{t \in TR(i)} n_t$.

We note that the set $TR(i)$ does not depend on the user, and that the user and item components of the factorization are fully uncoupled. This has the advantage that tag factors can also be exploited in the rating predictions for new users for whom tagging information is not available yet, improving over the standard matrix factorization method. The ItemRelTags model, however, does not take into account the possibility that the user has tagged different items other than the one for which the rating is being estimated. In such cases, it may be beneficial to enrich the user's profile by considering other tags the user has chosen in the past as evidence of her preferences. In the next subsection, we propose a model that aims to exploit this information to generate more accurate recommendations.

Similarly to the SVD++ algorithm, all of the above models can be trained by minimizing the associated loss function with stochastic gradient descent.

4.2 Adaptation of gSVD++ for Tag-based Cross-domain Collaborative Filtering

Although the previous recommendation models can successfully transfer tagging information between domains, they suffer from some limitations. The UserItemTags and UserItemRelTags models cannot do better than the standard matrix factorization if the user has not tagged the item for which the rating is being estimated, while the ItemRelTags model does not fully exploits the user's preferences expressed in the tags assigned to other items.

In this paper, we propose to adapt the gSVD++ algorithm by introducing an additional set of latent variables $x_s \in \mathbb{R}^k$ that enrich the user's factors and better capture the effect of her tags in the rating estimation. Specifically, we distinguish between two different sets of tags for users and items, and factorize the rating matrix into fully uncoupled user and item components as follows:

$$\hat{r}_{ui} = \left(p_u + \frac{1}{|T_u|} \sum_{s \in T_u} n_{us} x_s \right)^T \left(q_i + \frac{1}{|T_i|} \sum_{t \in T_i} n_{it} y_t \right) \tag{11}$$

The set T_u contains all the tags assigned by user u to any item. Respectively, T_i is the set of tags assigned by any user to item i, and plays the role of item metadata $G(i)$ in the gSVD++ algorithm. As in the ItemRelTags model, there may be repeated tags in each of the above tag sets, which we account for by considering the number of times a tag appears in T_u or T_i, respectively. In (11), n_{us} is the number of items on which the user u applied tag s, and n_{it} is the number of users that applied tag t to item i. As previously, tag factors are normalized by $|T_u| = \sum_{s \in T_u} n_{us}$ and $|T_i| = \sum_{t \in T_i} n_{it}$, so that factors x_s and y_t do not dominate over the rating factors p_u and q_i for users and items with a large number of tags.

In the proposed model, which we call as **TagGSVD++**, a user's profile is enhanced with the tags she used, since we hypothesize that her interests are better captured, and that transferring this information between domains can be beneficial for estimating ratings in the target domain. Likewise, item profiles are extended with the tags that were applied to them, as in the ItemRelTags model.

The parameters of TagGSVD++ can be learned from the observed training data by solving the following unconstrained minimization problem:

$$\min_{p_*,q_*,x_*,y_*} \sum_{(u,i)\in\mathcal{R}} E\left(p_u, q_i, \{x_s\}_{s\in T_u}, \{y_t\}_{t\in T_i}\right)$$
$$= \min_{p_*,q_*,x_*,y_*} \sum_{(u,i)\in\mathcal{R}} \frac{1}{2}\left[r_{ui} - \left(p_u + \frac{1}{|T_u|}\sum_{s\in T_u} n_s x_s\right)^T \left(q_i + \frac{1}{|T_i|}\sum_{t\in T_i} n_t y_t\right)\right]^2 \quad (12)$$
$$+ \frac{\lambda}{2}\left(\|p_u\|^2 + \|q_i\|^2 + \sum_{s\in T_u}\|x_s\|^2 + \sum_{t\in T_i}\|y_t\|^2\right)$$

The factor $1/2$ simplifies the following derivations with no effect on the solution. As in the previous models, a minimum can be found by stochastic gradient descent. For completeness, in the following we list the update rules of TagGSVD++ taking the derivatives of the error function in (12) with respect to the parameters:

$$\frac{\partial E}{\partial p_u} = -e_{ui}\left(q_i + \frac{1}{|T_i|}\sum_{t\in T_i} n_{it} y_t\right) + \lambda p_u$$
$$\frac{\partial E}{\partial q_i} = -e_{ui}\left(p_u + \frac{1}{|T_u|}\sum_{s\in T_u} n_{us} x_s\right) + \lambda q_i$$
$$\frac{\partial E}{\partial x_a} = -e_{ui}\frac{n_{ua}}{|T_u|}\left(q_i + \frac{1}{|T_i|}\sum_{t\in T_i} n_{it} y_t\right) + \lambda x_a \qquad \forall a \in T_u$$
$$\frac{\partial E}{\partial y_b} = -e_{ui}\frac{n_{ib}}{|T_i|}\left(p_u + \frac{1}{|T_u|}\sum_{s\in T_u} n_{us} x_s\right) + \lambda y_b \qquad \forall b \in T_i$$

where the error term e_{ui} is $r_{ui} - \hat{r}_{ui}$. In the training phase, we loop over the observed ratings simultaneously updating the parameters according to the following rules:

$$p_u \leftarrow p_u - \alpha\left[\lambda p_u - e_{ui}\left(q_i + \tfrac{1}{|T_i|}\textstyle\sum_{t\in T_i} n_{it} y_t\right)\right]$$
$$q_i \leftarrow q_i - \alpha\left[\lambda q_i - e_{ui}\left(p_u + \tfrac{1}{|T_u|}\textstyle\sum_{s\in T_u} n_{us} x_s\right)\right]$$
$$x_a \leftarrow x_a - \alpha\left[\lambda x_a - e_{ui}\tfrac{n_{ua}}{|T_u|}\left(q_i + \tfrac{1}{|T_i|}\textstyle\sum_{t\in T_i} n_{it} y_t\right)\right], \forall a \in T_u$$
$$y_b \leftarrow y_b - \alpha\left[\lambda y_b - e_{ui}\tfrac{n_{ib}}{|T_i|}\left(p_u + \tfrac{1}{|T_u|}\textstyle\sum_{s\in T_u} n_{us} x_s\right)\right], \forall b \in T_i$$

The learning rate α determines to what extent the parameters are updated in each iteration. A small learning rate can make the learning slow, whereas with a large learning rate the algorithm may fail to converge. The choice of both the learning rate and the regularization parameter λ is discussed later in section 5.3.

5. EXPERIMENTS

We have evaluated the proposed TagGSVD++ algorithm (section 4.2) in a cross-domain collaborative filtering setting, by empirically comparing it with the single-domain matrix factorization methods (section 3) and the state-of-the-art cross-domain recommendation approaches described in section 4.1.

5.1 Dataset

We have attempted to reproduce the cross-domain dataset used in [6], aiming to compare our approach with those presented in that paper. For the sake of completeness, we also describe the data collection process here.

In order to simulate the cross-domain collaborative filtering setting, we have downloaded two publicly available datasets for the movies and books domains. The MovieLens 10M dataset[6] (ML) contains over 10 million ratings and 100,000 tag assignments by 71,567 users to 10,681 movies. The LibraryThing dataset[7] (LT) contains over 700,000 ratings and 2 million tag assignments by 7,279 users on 37,232 books. Ratings in both of the datasets are expressed on a 1-5 scale, with interval steps of 0.5.

Since we were interested in analyzing the effect of tags on rating prediction, we only kept ratings in MovieLens on movies for which at least one tag was applied, leaving a total of 24,564 ratings. Also following the setup done by Enrich et al., we considered the same amount of ratings in LibraryThing, and took the first 24,564 ratings. We note, however, that the original dataset contained duplicate rows and inconsistencies, i.e., some user-item pairs had more than one rating. Hence, we preprocessed the dataset removing such repetitions and keeping only the repeated ratings that appeared first in the dataset's file. We also converted the tags to lower case in both datasets. Table 1 shows the characteristics of the final datasets.

Table 1. Details of the datasets used in the experiments after preprocessing.

	MovieLens	LibraryThing
Users	2,026	244
Items	5,088	12,801
Ratings	24,564	24,564
Avg. ratings per user	12.12	100.67
Rating sparsity	99.76%	99.21%
Tags	9,529	4,598
Tag assignments	44,805	72,943
Avg. tag assignments per user	22.16	298.95
Ratio of overlapping (shared) tags	13.81%	28.62%

5.2 Evaluation methodology

As mentioned above, we have compared the performance of the proposed model against the single-domain matrix factorization baselines from section 3, and the state-of-the-art tag-based algorithms described in section 4.1. All these methods are summarized next:

MF The standard matrix factorization method trained by stochastic gradient descent over the observed ratings of both movies and books domains.

SVD++ An adaptation of MF to take implicit data into account. In our experiments, the set $N(u)$ contains all the items rated by user u.

gSVD++ An extension of SVD++ to include item metadata into the factorization process. In our experiments, we have considered as set of item attributes $G(i)$ the tags T_i assigned to item i by any user. Note that, as tags are content features for both movies and books, this method is suitable for cross-domain recommendation, since knowledge can be transferred through the metadata (tag) factors. This differs from the proposed TagGSVD++ in that users are modeled as in SVD++ by considering rated items as implicit feedback instead of their tags. Also, normalization of the implicit data factors on the user component involves a square root; see equations (6) and (11).

UserItemTags A method that expands an item i's profile with latent factors of tags that the target user assigned to i. Its parameters are learned by simultaneously factorizing the rating matrices of both source and target domains.

UserItemRelTags A variation of the previous method that only takes relevant tags into account, as determined by a Wilcoxon rank-sum test.

[6] MovieLens datasets, http://grouplens.org/datasets/movielens
[7] LibraryThing dataset, http://www.macle.nl/tud/LT

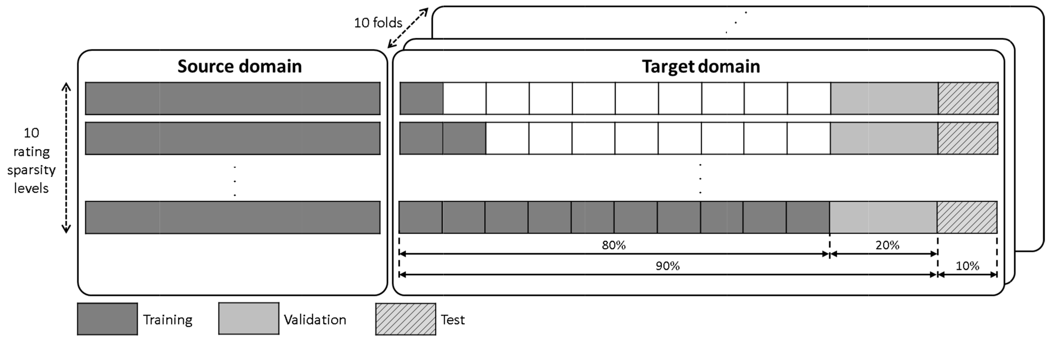


Figure 1. Data splitting done for cross-validation. Training data consists of source domain ratings and portions of the target domain, marked in dark.

ItemRelTags Instead of tags assigned by the user, this method exploits all relevant tags applied by the whole user community, and is thus able to compute rating predictions even if the user has not tagged the target item.

We evaluated all these recommendation methods in two settings, using MovieLens as source domain and LibraryThing as target domain, and vice-versa. In both cases, we evaluated the methods through 10-fold cross-validation, i.e., we shuffled the target ratings and split them into 10 non-overlapping folds. In each fold, we left out one part, 10% of the ratings, as *test* set to estimate the performance of the methods. The rest 90% of the ratings were used as a *training* set to learn the models, and a *validation* set to find the optimal values of the models' parameters. Specifically, we randomly chose 80% of these remaining ratings, and combined them with the source domain ratings to build the models. The final 20% left was used for the validation set to select the best number of factors k, learning rate α, and regularization λ. Figure 1 depicts the split of the data into training, validation, and test sets.

As in [6], we also wanted to investigate how the number of available ratings in the target domain affects the quality of the recommendations. For such purpose, we further split the training data from the target domain into 10 portions to simulate different rating sparsity levels. First, in order to evaluate the performance of the methods in cold-start situations, we used only 10% of the target training ratings, i.e., $0.1 \cdot 0.8 \cdot 0.9 \cdot 24{,}564 = 1{,}768$ ratings (see Table 1). Then, we incrementally added additional 10% of the ratings to analyze the behavior of the methods with an increasingly larger amount of observed rating data. In each sparsity level, the full set of source domain ratings was also used to build the models.

Since all the methods are designed for the rating prediction task, we measured their performance as the accuracy of the estimated ratings. Specifically, we computed the Mean Absolute Error (MAE) of each model in the different settings described above:

$$MAE = \frac{1}{|\mathcal{R}_{te}|} \sum_{(u,i) \in \mathcal{R}_{te}} |r_{ui} - \hat{r}_{ui}|$$

where $\mathcal{R}_{te}$ contains the ratings in the test set we left out for evaluation.

5.3 Results

As previously mentioned, we reserved 20% of the target domain training data in each fold for validating the models and finding the best model parameters, in order not to overestimate the performance of the methods.

For hyperparameter optimization, with each method and sparsity level in the target domain, we performed a grid (stepsize) search on the validation set for the values of the learning rate α, the amount of regularization λ, and the number of latent features k. To get an idea of the typical values found for the parameters, Table 2 shows the average best values for each method.

Table 2. Average values of the best parameters found.

	ML → LT			LT → ML		
	k	α	λ	k	α	λ
MF	41	0.020	0.009	43	0.020	0.009
SVD++	41	0.020	0.007	43	0.020	0.006
gSVD++	43	0.019	0.001	43	0.020	0.004
UserItemTags	46	0.019	0.003	46	0.020	0.010
UserItemRelTags	39	0.017	0.008	41	0.020	0.017
ItemRelTags	40	0.017	0.001	46	0.020	0.006
TagGSVD++	40	0.013	0.036	46	0.019	0.045

From the table, we observe that there is not a large difference in the optimal number of factors and learning rates between configurations. In contrast, we note that the amount of regularization needed for the proposed TagGSVD++ method is relatively large, e.g. compare $\lambda = 0.036$ of TagGSVD++ with $\lambda = 0.009$ of MF. This may be due to the additional set of latent variables for tags that our model uses; more complex models are able to account for greater variance in the data and tend to overfit more easily, thus requiring more regularization. In order to analyze how the available information in the target domain affects the stability of the model, Figure 2 shows the optimal value for the regularization parameter for different sparsity levels.

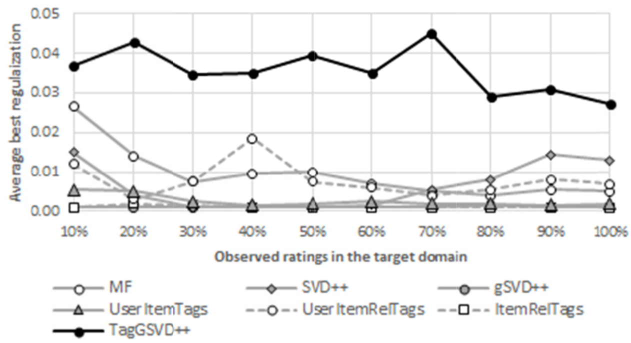
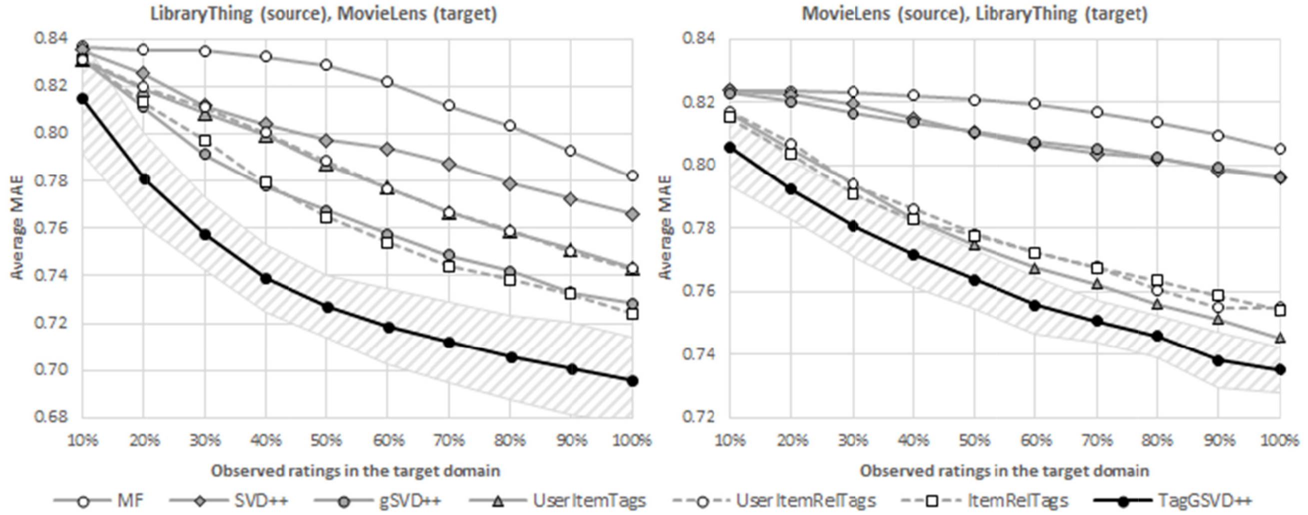


Figure 2. Optimal values for the regularization parameter using MovieLens as source domain and LibraryThing as target domain.



(a) **(b)**

Figure 3. Average prediction error over the 10 folds for different amounts of observed ratings in the target domain. The striped area represents the range of values within two standard deviations from the mean. (a) Results using LibraryThing as source domain and MovieLens as target domain. (b) Results using MovieLens as source domain, and LibraryThing as target domain.

We note that the gSVD++, upon which our model is defined, also introduces additional latent variables and yet requires a lower regularization. We argue that the differences between gSVD++ and TagGSVD++ regularizations are caused by the $N(u)$ and T_u sets, see equations (6) with $G(i) = T_i$ and (11). In Table 1 we see that, on average, the number of tags applied by a user is much larger than the number of rated items. This results in more variables actually taking part in the rating predictions, and hence in a more complex model that requires more regularization to prevent overfitting.

Once we found the best parameters for each method and sparsity level, we ran the models separately in the test set of each fold. The final performance was estimated as the average MAE over the 10 folds. Figure 3a shows the results obtained using LibraryThing as source domain and MovieLens as target domain. All the differences with respect to the TagGSVD++ algorithm are statistically significant as determined with a Wilcoxon signed rank test at the 95% confidence level. It can be seen that the proposed TagGSVD++ method is able to consistently outperform the rest of the methods for all sparsity levels in the target domain, also in the cold-start setting when only 10%-20% of the ratings are available. We also note that cross-domain methods always achieve better accuracy than single-domain MF, although SVD++ effectively exploits implicit feedback and remains competitive until the 50% sparsity level. Then, as the sparsity decreases, cross-domain models provide greater improvements. This indicates that even if plenty of target domain rating data is available, it is still beneficial to transfer knowledge from the source domain.

The results using MovieLens as source domain and LibraryThing as target domain are shown in Figure 3b. As before, the difference in MAE between TagGSVD++ and the rest of the methods is statistically significant, according to the Wilcoxon signed rank test with 95% confidence level. Again, TagGSVD++ is the best performing method for all rating sparsity levels, followed by the cross-domain methods. We now observe that the values of MAE are in general larger than in the previous case, which seems to indicate that the transfer of knowledge is not as effective in this setting. This observation is in accordance with the results reported in [6], where the authors argue that this may be caused by differences in the ratio of overlapping tags between the domains. Only 13.81% of the tags in MovieLens are shared in LibraryThing (see Table 1), and thus less latent tag factors learned in the source domain can be used in the target to compute rating predictions.

Figure 4 shows the average rating prediction error for users with different amounts of observed ratings and tag assignments, using LibraryThing as source domain and MovieLens as target domain. We see that our model achieves the best improvements in cold-start situations, where few ratings and tag assignments are available in the target domain. We also note that the performance degrades for users with more than 20 ratings (respectively, 100 tag assignments), when enough target domain data is available. Nonetheless, in these cases, TagGSVD++ is still able to exploit the learned tag factors to compute more accurate predictions.

6. CONCLUSIONS AND FUTURE WORK

Cross-domain recommendation has potential benefits over traditional recommender systems that focus on single domains, such as alleviating rating sparsity in a target domain by exploiting data from a related source domain, improving the quality of recommendations in cold-start situations by inferring new user preferences from other domains, and by personalizing cross-selling strategies to provide customers with suggestions of items of different types.

Despite these advantages, cross-domain recommendation is a fairly new topic with plenty of research opportunities to explore. One of the major difficulties that arises is how to link or relate the different domains to support the transfer of knowledge. Due to the common heterogeneity of item attributes across domains, collaborative filtering techniques have become more popular than content-based methods. However, recent work [6][17] has concluded that more reliable and meaningful relations can be established between the domains by exploiting certain content information, such as social tags.

In this paper, we have adapted a novel extension of the well-known SVD++ algorithm to separately model the effect of user and item tags in the observed ratings. By introducing a new set of latent variables that represent tags in the user profile, our TagGSVD++ method is able to transfer knowledge from a source domain more effectively, providing accurate rating predictions in

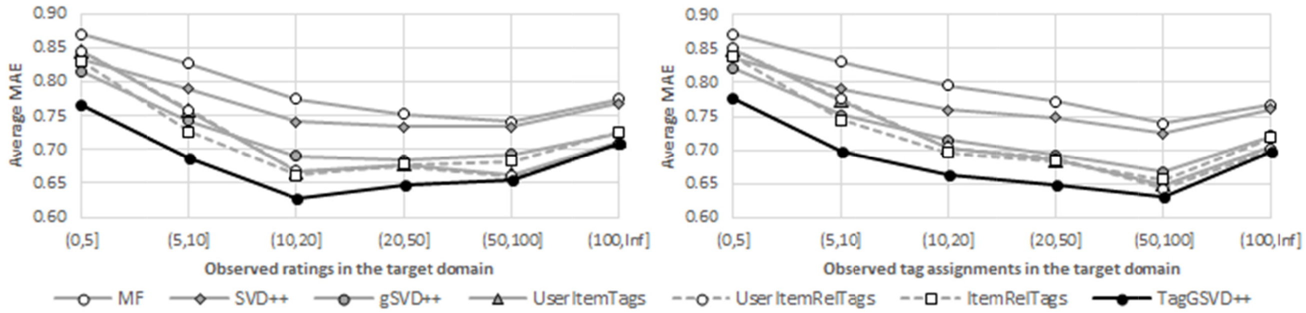


Figure 4. Average rating prediction error for users with different amounts of observed ratings (left) and tag assignments (right), using LibraryThing as source domain and MovieLens as target domain.

the target domain, even in cold-start situations. From our experiments in the movies and books domains, we conclude that exploiting additional tag factors, and decoupling user and item components in the factorization process improves the transfer of knowledge and the accuracy of the recommendations.

In the future, we plan to further investigate the effect of tags in the quality of recommendations. In particular, we want to study how the recommendation performance depends on the number of shared tags between domains. Increasing the overlap by grouping tags with similar semantics but expressed differently in the domains could favor the transfer of knowledge.

In our experiments we altered the amount of observed rating data in the target domain, but it would also be interesting to evaluate the methods varying the number of available ratings in the source domain. Moreover, we will perform a more exhaustive evaluation with other datasets including more cross-domain recommendation methods from the state of the art, such as [17].

7. ACKNOWLEDGEMENTS

This work was supported by the Spanish Government (TIN2013-47090-C3-2).

8. REFERENCES

[1] Abel, F., Helder, E., Houben, G.-J., Henze, N., Krause, D. 2013. Cross-system User Modeling and Personalization on the Social Web. *User Modeling and User-Adapted Interaction* 23(2-3), pp. 169-209.

[2] Adomavicius, G., Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, pp. 734–749.

[3] Berkovsky, S., Kuflik, T., Ricci, F. 2008. Mediation of User Models for Enhanced Personalization in Recommender Systems. *User Modeling and User-Adapted Interaction* 18(3), pp. 245-286.

[4] Cao, B., Liu, N. N., Yang, Q. 2010. Transfer Learning for Collective Link Prediction in Multiple Heterogeneous Domains. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 159-166.

[5] Cremonesi, P., Tripodi, A., Turrin, R. 2011. Cross-domain Recommender Systems. In *Proceedings of the 11th IEEE International Conference on Data Mining Workshops*, pp. 496-503.

[6] Enrich, M., Braunhofer, M., Ricci, F. 2013. Cold-Start Management with Cross-Domain Collaborative Filtering and Tags. In *Proceedings of the 14th International Conference on E-Commerce and Web Technologies,* pp. 101-112.

[7] Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F. 2012. Cross-domain Recommender Systems: A Survey of the State of the Art. In *Proceedings of the 2nd Spanish Conference on Information Retrieval*, pp. 187-198.

[8] Funk, S. 2006. Netflix Update: Try This At Home. http://sifter.org/~simon/journal/20061211.html

[9] Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J. 2013. Cross-Domain Recommendation via Cluster-Level Latent Factor Model. In *Proceedings of the 17th and 24th European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 161-176.

[10] Garcia Manzato, M. 2013. gSVD++: Supporting Implicit Feedback on Recommender Systems with Metadata Awareness. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp. 908-913.

[11] Koren, Y. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426-434.

[12] Koren, Y., Bell, R., Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42(8), pp. 30-37.

[13] Koren, Y., Bell, R. 2011. Advances in Collaborative Filtering. *Recommender Systems Handbook*, pp. 145-186.

[14] Li, B., Yang, Q., Xue, X. 2009. Can Movies and Books Collaborate? Cross-domain Collaborative Filtering for Sparsity Reduction. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 2052-2057.

[15] Pan, W., Xiang, E.W., Liu, N.N., Yang, Q. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 210-235.

[16] Shapira, B., Rokach, L., Freilikhman, S. 2013. Facebook Single and Cross Domain Data for Recommendation Systems. *User Modeling and User-Adapted Interaction* 23(2-3), pp. 211-247.

[17] Shi, Y., Larson, M., Hanjalic, A. 2011. Tags as Bridges between Domains: Improving Recommendation with Tag-induced Cross-domain Collaborative Filtering. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, pp. 305-316.

[18] Tiroshi, A., Berkovsky, S., Kaafar, M. A., Chen, T., Kuflik, T. 2013. Cross Social Networks Interests Predictions Based on Graph Features. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 319-322.

[19] Winoto, P., Tang, T. 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, pp. 209-225.

[20] Zhang, Y., Cao, B., Yeung, D.-Y. 2010. Multi-Domain Collaborative Filtering. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pp. 725–732.