



Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization

Ignacio Fernández-Tobías¹ · Iván Cantador¹ · Paolo Tomeo² · Vito Walter Anelli² · Tommaso Di Noia²

Received: 15 December 2017 / Accepted in revised form: 18 December 2018 / Published online: 1 January 2019
© Springer Nature B.V. 2019

Abstract

Providing relevant personalized recommendations for new users is one of the major challenges in recommender systems. This problem, known as the user *cold start* has been approached from different perspectives. In particular, cross-domain recommendation methods exploit data from source domains to address the lack of user preferences in a target domain. Most of the cross-domain approaches proposed so far follow the paradigm of collaborative filtering, and avoid analyzing the contents of the items, which are usually highly heterogeneous in the cross-domain setting. Content-based filtering, however, has been successfully applied in domains where item content and metadata play a key role. Such domains are not limited to scenarios where items do have text contents (e.g., books, news articles, scientific papers, and web pages), and where text mining and information retrieval techniques are often used. Potential application domains include those where items have associated metadata, e.g., genres, directors and actors for movies, and music styles, composers and themes for songs. With the advent of the Semantic Web, and its reference implementation Linked Data, a plethora of structured, interlinked metadata is available on the Web. These metadata represent a potential source of information to be exploited by content-based and hybrid filtering approaches. Motivated by the use of Linked Data for recommendation purposes, in this paper we present and evaluate a number of matrix factorization models for cross-domain collaborative filtering that leverage metadata as a bridge between items liked by users in different domains. We show that in case the underlying knowledge graph connects items from different domains and then in situations that benefit from cross-domain information, our models can provide better recommendations to new users while keeping a good trade-off between recommendation accuracy and diversity.

Keywords Cross-domain recommender systems · User cold start · Item metadata · Linked data

Extended author information available on the last page of the article

1 Introduction

Cross-domain recommendation has recently emerged as a potential solution to the cold start problem in recommender systems (Cantador et al. 2015), aiming to mitigate the lack of data by exploiting user preferences and item attributes in domains distinct but related to the target domain. In this line, most of the cross-domain approaches proposed so far are based on collaborative filtering (Cremonesi et al. 2011), exploiting user preferences as a bridge to relate source and target domains, and ignoring the content of the items. Hence, they benefit from the fact that they do not need to perform any kind of analysis of item contents, which are in general highly heterogeneous across domains, and whose inter-domain relationships may be difficult to establish.

These difficulties, however, can be addressed nowadays thanks to the Semantic Web initiative (Shadbolt et al. 2006), and more specifically to its reference implementation the Linked Open Data (LOD) project (Bizer et al. 2009), which has originated a large number of inter-linked knowledge repositories publicly available in the Web, following the Semantic Web standards for data representation and access. Hence, in the current Web there is a wide array of structured data sources with information of items belonging to a variety of domains, such as history, arts, science, industry, media and sports, to name a few. This information not only consists of particular multimedia contents and associated metadata, but also explicit, semantic relations between items and metadata.

Motivated by the availability of large amounts of item metadata and semantic relations in the Linked Data cloud, we aim to address the cross-domain recommendation problem not only focusing on user preferences and item attributes, but also exploiting content-based relations between items from different domains. More specifically, we propose to use the set of LOD semantic features and relations as inter-domain links for supporting knowledge transfer across domains, enabling cross-domain item similarities, and providing recommendations for cold start users in the target domain.

Previous work has proposed graph-based algorithms to address the recommendation problem in heterogeneous datasets (Yu et al. 2014; Di Noia et al. 2016), analyzing the topology of semantic networks to jointly exploit user preferences and item metadata. These approaches have been shown to be effective for recommendation, but suffer from computational issues caused by the size of the semantic networks, which are in general very large. Differently, we avoid these issues by working in two steps. First, we exploit the semantic networks to compute inter-domain similarities that link items from different domains. Then, we leverage the computed similarities in hybrid matrix factorization (MF) models for recommendation, which no longer need to deal with the whole networks.

Therefore, the main contribution of this work is the development of novel, effective hybrid matrix factorization models that jointly exploit user preferences and item metadata for cross-domain recommendation. Moreover, we adapt a fast learning algorithm by Pilászy et al. (2010) for efficiently building our models, and

evaluate them in cold start scenarios in several domains, in terms of both precision and diversity.

We evaluate the performance of the proposed models using a dataset of Facebook¹ *likes* about movies, music and books. In order to obtain semantic metadata for the different items, we first mapped the items in our dataset to entities in LOD by means of SPARQL queries, and then extracted their attributes and relations to enhance the item profiles.

In a first experiment, we compared several state-of-the-art semantic similarity metrics for content-based recommendation, aiming to understand which is more suitable for later injecting in our cross-domain MF models, and achieving the best results using the link-based approach by Milne and Witten (2008). Then, we evaluated the ranking precision and diversity of the recommendations computed by the proposed models. We show that, depending on the involved source and target domains, our models generate more accurate suggestions than a number of baselines in severe cold start situations. Moreover, the proposed approaches provide a better trade-off between accuracy and diversity, which are in general difficult to balance.

We point out that the presented approaches can be effectively used if the underlying LOD knowledge graph encodes direct or indirect connections between items in different domains. In fact, we need to compute semantic similarities between items not belonging to the same domain. These connections are quite common for knowledge domains with some degree of information overlap, such as in the case of books, movies and music but, in case they are missing or rare, this may result as a limitation for the performances of the approaches we introduce here.

The reminder of the paper is structured as follows. In Sect. 2, we revise related work on cross-domain recommender systems, focusing on those approaches that are based on Matrix Factorization. For the sake of completeness, we provide an overview of the standard Matrix Factorization technique in Sect. 3. In Sect. 4, we present the developed cross-domain hybrid matrix factorization models. Next, in Sect. 5, we report and analyze the empirical results achieved in the experiments conducted to analyze user cold start situations. Finally, in Sect. 6 we end with some conclusions and future research lines.

2 Related work

In this section, we survey the state of the art on cross-domain recommender systems. First, in Sect. 2.1 we describe the cross-domain recommendation problem and present a categorization of the approaches, giving representative examples of each category. Next, in Sect. 2.2 we focus on those cross-domain recommendation approaches that use the matrix factorization technique to bridge the source and target domains.

¹ Facebook online social networking, <https://www.facebook.com>.

2.1 Cross-domain recommender systems

Nowadays, the majority of recommender systems suggest items belonging to a single domain. For instance, Netflix² recommends movies and TV shows, Spotify³ recommends songs and music albums, and Barnes & Noble⁴ recommends books. These domain-specific systems have been successfully deployed by numerous web platforms, and the single-domain recommendation functionality is not perceived as a limitation, but rather pitched as a focus on a certain market.

Nonetheless, in large e-commerce sites such as Amazon.com⁵ and eBay⁶ users often provide feedback for items from multiple domains, and in social networks like Facebook⁷ and Twitter⁸ users express their tastes and interests for a variety of topics. It may, therefore, be beneficial to leverage all the available user data provided in various systems and domains in order to generate more encompassing user models and better recommendations. Instead of treating each domain (e.g., movies, music and books) independently, knowledge acquired in a *source* domain could be transferred to and exploited in another *target* domain. The research challenge of transferring knowledge, and the business potential of delivering recommendations spanning across multiple domains, have triggered an increasing interest in *cross-domain recommendations*.

The cross-domain recommendation problem has been addressed from various perspectives in different research areas. It has been handled by means of user preference aggregation and mediation strategies for cross-system personalization in User Modeling (Abel et al. 2013; Berkovsky et al. 2008b; Shapira et al. 2013), as a potential solution to mitigate the cold start and sparsity problems in Recommender Systems (Cremonesi et al. 2011; Shi et al. 2011; Tiroshi et al. 2013), and as a practical application of knowledge transfer in Machine Learning (Gao et al. 2013; Li et al. 2009a; Pan et al. 2010). Focusing on how knowledge is exploited by cross-domain recommender systems, in Cantador et al. (2015) we categorized existing works according a two-level taxonomy.

- *Aggregating knowledge* Knowledge from various source domains is aggregated to perform recommendations in a target domain. Depending on the stage in the recommendation process where the aggregation is performed we can further distinguish three cases. First, we find approaches that *merge user preferences* e.g., ratings, tags, transaction logs, and click-through data. The aggregation can be done by means of a multi-domain rating matrix (Berkovsky et al. 2007a; Sahebi and Brusilovsky 2013), using a common representation for user preferences such as social tags (Szomszor et al. 2008a; Abel et al. 2013; Fernández-Tobías et al. 2013) or semantic concepts (Kaminskas et al. 2013), linking the preferences via a multi-domain graph (Cremonesi et al. 2011; Tiroshi et al. 2013), or mapping user preferences to

² Netflix streaming media and video provider, <https://www.netflix.com>.

³ Spotify digital music service, <https://www.spotify.com>.

⁴ Barnes & Noble online bookseller, <http://www.barnesandnoble.com>.

⁵ Amazon electronic commerce site, <https://www.amazon.com>.

⁶ eBay consumer-to-consumer and business-to-consumer sales, <http://www.ebay.com>.

⁷ Facebook social network, <https://www.facebook.com>.

⁸ Twitter online news and social networking service, <https://twitter.com>.

domain-independent features such as personality traits (Cantador et al. 2013) or user–item interaction features (Loni et al. 2014). In the second case, user modeling data from various recommender systems is *mediated* to improve target recommendations. For instance, Berkovsky et al. (2007a), Tiroshi and Kuflik (2012) and Shapira et al. (2013) import user neighborhoods and user–user similarities computed in the source domain into the target. Finally, some approaches directly *combine* single-domain recommendations, e.g. rating estimations (Berkovsky et al. 2007a; Givon and Lavrenko 2009) and rating probability distributions (Zhuang et al. 2010).

- *Linking and transferring knowledge* Knowledge linkage or transfer between domains is established to support recommendations. In this case, we find methods that (1) *link domains* by a common knowledge such as item attributes (Chung et al. 2007), association rules (Cantador et al. 2013), semantic networks (Fernández-Tobías et al. 2011; Kaminskis et al. 2013), and inter-domain correlations (Zhang et al. 2010; Shi et al. 2011; Sahebi et al. 2017); methods that (2) *share latent features* between source and target domains factor models, either by using same model parameters (Pan et al. 2011; Hu et al. 2013; He et al. 2018) in both factorizations, or by introducing new parameters that extend the factorizations (Enrich et al. 2013; Fernández-Tobías and Cantador 2014); and methods that (3) *transfer rating patterns* extracted by co-clustering the source domain rating matrix and exploit them in the target domain (Li et al. 2009a; Gao et al. 2013; Cremonesi and Quadran 2014). After defining the problem, in Pan (2016) three different knowledge transfer strategies for collaborative recommendation with auxiliary data (TL-CRAD) are introduced: (1) adaptive knowledge transfer, (2) collective knowledge transfer and (3) integrative knowledge transfer. Then, for each of them the author surveys related work with reference to different knowledge strategies with an emphasis on: transfer via prediction rule, transfer via regularization and transfer via constraint.

In terms of the goals addressed by cross-domain recommenders, we find great diversity among the reviewed approaches. Most proposals focus on improving accuracy by reducing data sparsity (Li et al. 2009a; Shi et al. 2011; Cao et al. 2010; Zhang et al. 2010; Pan et al. 2010; Tiroshi et al. 2013; Loni et al. 2014; Zhu et al. 2018). In many domains, the average number of ratings per user and item is low, which may negatively affect the quality of the recommendations. Data collected outside the target domain can increase the rating density, and thus may upgrade the recommendation quality. Others seek to enhance user models, which may have personalization-oriented benefits such as (1) discovering new user preferences for the target domain (Stewart et al. 2009; Szomszor et al. 2008b), (2) enhancing similarities between users and items (Abel et al. 2011; Berkovsky et al. 2008b), and (3) measuring vulnerability in social networks (Goga et al. 2013; Jain et al. 2013). Cross-domain methods have been also applied to bootstrap recommender systems by importing preferences from another source outside the target domain (Shapira et al. 2013), and have been proposed to improve the diversity of recommendations by providing better coverage of the range of user preferences (Winoto and Tang 2008). Finally, a few approaches have dealt with the new user problem (Hu et al. 2013; Sahebi and Brusilovsky 2013; Tiroshi and Kuflik 2012; Enrich et al. 2013). When a user starts using a recommender system, it

has no knowledge of the user's tastes and interests, and cannot produce personalized recommendations. This may be solved by exploiting the user's preferences collected in a different source domain.

We observe that addressing the cold-start has been barely investigated, as in Liu et al. (2017) where the authors present a neighborhood-based algorithm for the dual cold-start problem. The generalization of users and items into a cluster level to obtain high-quality relations also in cold start scenario is the focus of Mirbakhsh and Ling (2015). They first employ biased matrix factorization to map rating matrix into lower-dimension latent spaces. After this step, they apply the k-means clustering algorithm to categorize users and items. The cold-start is also the main topic of Wongchokprasitti et al. (2015), where the authors propose a novel approach to cross-system personalization based on two assumptions: the existence of a user model that could be shared among platforms, and that a specific system can maintain (and provide) the user models built by its system.

As we shall present in Sect. 4, we aim to deal with the cold-start problem by means of novel matrix factorization models that jointly exploit user ratings and item metadata. Before, in Sect. 2.2, we revise state-of-the-art cross-domain recommender systems based on matrix factorization.

2.2 Matrix factorization-based cross-domain recommender systems

Although matrix factorization models can be applied in cross-domain approaches based on knowledge aggregation—essentially as a standard recommendation problem once the user preferences from both domains are combined—, they have been mostly used in knowledge linkage and transfer approaches. In these settings, latent factors from source and target domains are either shared or related in order to establish the bridge between the domains.

One way of linking domains explored in previous works exploits inter-domain similarities by integrating them into the probabilistic matrix factorization model (Salakhutdinov and Mnih 2007). Specifically, such similarities are imposed as constraints over user or item latent factors when jointly factorizing rating matrices. For instance, Cao et al. (2010) proposed an approach in which inter-domain similarities are implicitly learned from data, as model parameters in a non-parametric Bayesian framework. Since user feedback is used to estimate the similarities, user overlap between the domains is required. Addressing the sparsity problem, Zhang et al. (2010) adapted the probabilistic matrix factorization method to include a probability distribution of user latent factors that encodes inter-domain correlations. One strength of this approach is that user latent factors shared across domains are not needed, allowing more flexibility in capturing the heterogeneity of domains. Instead of automatically learning implicit correlations in the data, Shi et al. (2011) argued that explicit common information is more effective, and relied on shared social tags to compute cross-domain user-to-user and item-to-item similarities. Similarly to previous approaches, rating matrices from the source and target domains are jointly factorized; but in this case user, and item latent factors from each domain are restricted, so that their product is consistent with the tag-based similarities.

Latent factors shared between domains can be exploited to support cross-domain recommendations. In this context, two types of approaches have been studied to perform the actual transfer of knowledge; namely, *adaptive* and *collective* models. In the former, latent factors are learned in the source domain, and are integrated into a recommendation model in the target domain, while in the latter, latent factors are learned simultaneously optimizing an objective function that involves both domains. Pan et al. (2010) addressed the sparsity problem in the target domain following the adaptive approach, proposing to exploit user and item information from auxiliary domains where user feedback may be represented differently. In particular, they studied the case in which users express binary like/dislike preferences in the source domain, and utilize 1–5 ratings in the target domain. Their approach performs singular value decomposition (SVD) in each auxiliary domain in order to separately compute user and item latent factors, which are then shared with the target domain. Specifically, transferred factors are integrated into the factorization of the rating matrix in the target domain, and added as regularization terms so that specific characteristics of the target domain can be captured. Latent factors can also be shared in a collective way, as studied by Pan et al. (2011). In this case, instead of learning latent features from the source domains and transferring them to the target domain, the authors proposed to learn the latent features simultaneously in all the domains. Both user and item factors are assumed to generate the observed ratings in every domain, and, thus, their corresponding random variables are shared between the probabilistic factorization models of each rating matrix. Moreover, the factorization method is further extended by incorporating another set of factors that capture domain-dependent information, resulting in a tri-factorization scheme. A limitation of the proposed approach is that the users and items from the source and target domains have to be identical. Instead of focusing on sharing latent factors, Enrich et al. (2013) and Fernández-Tobías and Cantador (2014) studied the influence of social tags on rating prediction, as a knowledge transfer approach for cross-domain recommendations. The authors presented a number of models based on the SVD++ algorithm (Koren 2008) to incorporate the effect of tag assignments into rating estimation. The underlying hypothesis is that information about item annotation in a source domain can be exploited to improve rating prediction in a target domain, as long as a set of common tags between the domains exists. In the proposed models, tag factors are added to the latent item vectors, and are combined with user latent features to compute rating estimations. The difference between these models is in the set of tags considered for rating prediction. In all the models, knowledge transfer is performed through the shared tag factors in a collective way, since these are computed jointly for the source and the target domains. Hu et al. (2013) presented a more complex approach that takes domain factors into account. There, the authors argue that user–item dyadic data cannot fully capture the heterogeneity of items, and that modeling domain-specific information is essential to make accurate predictions in a setting where users typically express their preferences in a single domain. They referred to this problem as the *unacquainted world*, and proposed a tensor factorization algorithm to exploit the triadic user–item–domain data. In that method, rating matrices from several domains are simultaneously decomposed into shared user, item, and domain latent factors, and a genetic algorithm automatically estimates optimal weights of the domains. In a recent work, Zhu et al. (2018) propose

a two-step approach where the latent factors, learned via MF for both the source and target domains, are linked by training a deep neural network (DNN) representing their connections. Interestingly, the training process of the DNN is driven by the sparsity degrees of individual users and items in the source and target domains. Contextual and content-based information is exploited in Taneja and Arora (2018) to cluster users in the source domain prior to a tensor factorization. The proposed Cross Domain-Multi Dimensional Tensor Factorization (CD-MDTF) mitigates the sparsity and cold-start problem by transferring the aggregated knowledge from the source domain to target domain. An approach based on linking and transferring knowledge is proposed in Zhao et al. (2017), where the main assumption is that correspondences among entities in different domains are unknown, but can be computed with a cost. Starting from this assumption, the authors propose a unified framework aimed at actively mapping entities in different domain and then transferring knowledge via collaborative filtering. This latter step leverages partial mappings among entities for knowledge transfer. The authors also show how to integrate in their framework various extended matrix factorization techniques in a transfer learning manner. An emphasis on the meaningfulness of the knowledge extracted from the source domain to the target domain is the main topic in Zhang et al. (2017). A clustering step among users and items is performed both in the source and target before a matrix factorization. Then, by comparing the resulting matrices, it is possible to evaluate the consistency of the information transfer.

Rather than sharing user or item latent factors for knowledge transfer, a different set of approaches analyzes the structure of rating data at the community level. These methods are based on the hypothesis that even when their users and items are different, close domains are likely to have user preferences sampled with the same population. Therefore, latent correlations may exist between preferences of groups of users for groups of items, which are referred to as *rating patterns*. In this context, rating patterns can act as a bridge that relates the domains, such that knowledge transfer can be performed in either adaptive or collective manners. In the adaptive setting, rating patterns are extracted from a dense source domain (Li et al. 2009a; Gao et al. 2013). In the collective setting, data from all the domains are pulled together and jointly exploited, even though users and items do not overlap across domains (Li et al. 2009b). In He et al. (2018), the authors propose to alleviate the data sparsity problem in a target domain by transferring rating patterns from multiple incomplete source domains. The proposed approach extracts rating patterns from a sparse source domain that are eventually combined with collaborative filtering to approximate the target domain and predict missing values. In particular, they take into account the effects related to negative transfer to obtain more robust recommendations.

3 Matrix factorization-based collaborative filtering

Matrix Factorization (MF) models are among the most popular approaches to collaborative filtering, and have been actively investigated since they were introduced in the context of the Netflix prize competition (Bell and Koren 2007a). As opposed to classic user- and item-based collaborative filtering heuristics (Herlocker et al. 1999; Linden et al. 2003), MF methods train a statistical model from the available data using

machine learning techniques. Specifically, they perform a *dimensionality reduction* of the highly sparse rating matrix into a subspace of latent factors, which aim to capture implicit properties of users and items. In order for MF to be effective, the dimension k of the latent subspace is assumed to be much smaller than the number of users and items, $k \ll \min(|U|, |I|)$, essentially acting as a *bottleneck* that compresses the sparse input while retaining enough information to explain the observed user–item interactions.

In this section we review the classical matrix factorization models for explicit and implicit feedback, which serve as the building blocks for our proposed models presented in Sect. 4, as well as the state-of-the-art approaches BPR and FISM that we use as baselines in our experiments.

3.1 Matrix factorization models for rating prediction

Recommendation models based on MF have their roots on the *Latent Semantic Analysis* (LSA) technique (Deerwester et al. 1990), widely used in Natural Language Processing and Information Retrieval. LSA attempts to automatically infer concepts implicit in text documents by approximating the term–document matrix with a truncated Singular Value Decomposition (SVD) of lower rank. The first MF approaches for recommendation borrowed the same idea, and applied it to the user–item matrix in the rating prediction task (Sarwar et al. 2000). In contrast to LSA, the SVD is not well defined for sparse matrices as those commonly found in recommender systems, and hence the above approaches relied on imputation techniques to fill the missing matrix entries before applying SVD.

Rather than filling the rating matrix, which may introduce inaccurate information, subsequent approaches aimed to only factorize observed ratings instead of the whole matrix. One of the first and most popular methods in this line is the model proposed by Funk (2006), in which each user u is assigned a vector $\mathbf{p}_u \in \mathbb{R}^k$ of latent features automatically inferred from the data, and similarly each item i is assigned a vector $\mathbf{q}_i \in \mathbb{R}^k$ in the same subspace. Intuitively, latent features aim to capture properties implicit in the data—such as the amount of *comedy* or *action* in the case of movies—, but does not need to be interpretable at all, as this is not enforced in the model (Koren and Bell 2015). Ratings are then estimated as the dot product of latent feature vectors:

$$\hat{r}(u, i) = \langle \mathbf{p}_u, \mathbf{q}_i \rangle \quad (1)$$

Equivalently, the rating matrix \mathbf{R} is factorized as $\mathbf{R} \approx \mathbf{P}\mathbf{Q}^\top$, where \mathbf{P} is a $|U| \times k$ matrix with the user vectors \mathbf{p}_u as rows, and respectively \mathbf{Q} is $|I| \times k$ contains the \mathbf{q}_i as rows. The values of these matrices are automatically estimated from the data, by minimizing the Mean Squared Error of the ratings predicted against the ratings observed in a training set. That is, \mathbf{P} and \mathbf{Q} are chosen to minimize to following loss function:

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \sum_{(u,i) \in \mathcal{R}} \left\{ (r_{ui} - \langle \mathbf{p}_u, \mathbf{q}_i \rangle)^2 + \lambda (\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2) \right\} \quad (2)$$

where \mathcal{R} is the set of observed ratings, i.e., the set of non-zero entries of the rating matrix \mathbf{R} , and $\lambda > 0$ is a regularization hyper-parameter used to prevent overfitting.

In (Funk 2006) this function is minimized using *Stochastic Gradient Descent*, a widely used optimization technique that iteratively updates the parameters in the opposite direction of the gradient. When applied to Eq. (2), this technique yields the following update rules for the parameters \mathbf{p}_u and \mathbf{q}_i for each rating r_{ui} in the training set:

$$\mathbf{p}_u \leftarrow \mathbf{p}_u - \eta (e_{ui} \mathbf{q}_i + \lambda \mathbf{p}_u) \quad (3)$$

$$\mathbf{q}_i \leftarrow \mathbf{q}_i - \eta (e_{ui} \mathbf{p}_u + \lambda \mathbf{q}_i) \quad (4)$$

The *learning rate* η is a hyper-parameter that controls the extent to which the model parameters are updated in each iteration, and is carefully chosen; too large values may make the algorithm fail to converge, while too small values may make its convergence very slow. e_{ui} is the prediction error, and is defined as $e_{ui} \triangleq r_{ui} - \hat{r}(u, i)$.

In addition to Stochastic Gradient Descent, other optimization techniques have been explored in the literature, such as *Alternating Least Squares* (Bell and Koren 2007b), which is the standard technique followed in MF models for positive-only feedback (Sect. 3.2).

The basic SVD model by Funk (2006) is easily extensible, and has served as a building block for more complex matrix factorization models. For instance, Koren (2008) proposed the SVD++ model, which includes additional parameters to account for implicit feedback in rating predictions. Further extensions of SVD introduce temporal variables to capture the evolution of user preferences through time (Koren and Bell 2015).

3.2 Matrix factorization models for positive-only feedback

The core ideas behind the standard Matrix Factorization model for collaborative filtering have also been applied to the item ranking task when positive-only feedback is available instead of numeric ratings. Recommendation models designed for this type of data must take into account its particular characteristics, most notably the absence of negative feedback, but also the possible uncertainty in the positive feedback, as an observed user–item interaction may not necessarily indicate a preference of the user towards the item.

In one of the most representative works in this direction, Hu et al. (2008) proposed an adaptation of the rating-based MF model described previously to deal with positive-only feedback. As opposed to the rating-based SVD, which only considers the observed ratings, Hu et al.'s method models the full set of $|U| \cdot |I|$ interactions. Since negative feedback is not available in this scenario, the authors argue that the algorithm also has to model the missing information as an indirect source of negative user preferences. For such purpose, they introduce a parameter c_{ui} for each possible user–item pair that measures the confidence on the corresponding interaction, whether observed or not:

$$c_{ui} = 1 + \alpha k_{ui} \quad (5)$$

where k_{ui} is the count of implicitly collected interactions between user u and item i —such as number of clicks on a product web page on an e-commerce site, and the number of listening records of a given song in an online music provider—, and $\alpha > 0$ is a scaling parameter. When no interaction is observed, $k_{ui} = 0$ and the model assigns minimum confidence to the user–item pair, as it is unknown whether the lack of interaction is because the user really does not like the item, or just because the user does not know the item. Likewise, the more interactions are collected and the greater k_{ui} , the larger is the confidence on that observation. Moreover, focusing on the item ranking task, Hu et al.’s approach only aims to predict if the user will interact with the item, rather than the actual number of observations k_{ui} . Hence, a new set of variables is introduced so that $x_{ui} = 1$ if $k_{ui} > 0$, and $x_{ui} = 0$ otherwise.

Similarly to the SVD model for ratings, the recommendation score of item i for user u is estimated as the dot product of their corresponding latent feature vectors:

$$s(u, i) = \langle \mathbf{p}_u, \mathbf{q}_i \rangle \quad (6)$$

The model parameters \mathbf{p}_u and \mathbf{q}_i are again automatically learned by minimizing the mean squared error for the score predictions, but now accounting for the different confidence levels and the full set of possible user–item pairs:

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \sum_u \sum_i c_{ui} (x_{ui} - \langle \mathbf{p}_u, \mathbf{q}_i \rangle)^2 + \lambda (\|\mathbf{P}\|^2 + \|\mathbf{Q}\|^2) \quad (7)$$

Again, the loss function can be minimized with different numerical optimization techniques such as Stochastic Gradient Descent, but in Hu et al. (2008) the authors propose an *Alternating Least Squares* (ALS) procedure that efficiently handles the greater cost of accounting for the missing values. Clearly, the loss function in Eq. (7) involves many more terms than that of Eq. (2), as the number of observed entries in the user–item matrix is usually very small due to the data sparsity.

The key observation behind ALS is that when one set of parameters is fixed, the optimization problem in Eq. (7) is convex and analytically solvable using ordinary least-squares estimation. In particular, fixing the \mathbf{q}_i parameters and setting the gradient with respect to \mathbf{p}_u to zero yields the solution

$$\mathbf{p}_u = \left(\mathbf{Q}^\top \mathbf{C}^u \mathbf{Q} + \lambda \mathbf{I} \right)^{-1} \mathbf{Q}^\top \mathbf{C}^u \mathbf{x}_u . \quad (8)$$

where \mathbf{I} is the $k \times k$ identity matrix, \mathbf{C}^u is a $|I| \times |I|$ diagonal matrix with the c_{ui} values, and \mathbf{x}_u is a column vector of length $|I|$ with the x_{ui} values. The same procedure can be applied by fixing the user factors, and optimizing the item factors, leading to the solution

$$\mathbf{q}_i = \left(\mathbf{P}^\top \mathbf{C}^i \mathbf{P} + \lambda \mathbf{I} \right)^{-1} \mathbf{P}^\top \mathbf{C}^i \mathbf{x}_i . \quad (9)$$

Similarly, \mathbf{C}^i is a $|U| \times |U|$ diagonal matrix with the c_{ui} confidence values, and \mathbf{x}_i is a column vector of length $|U|$ containing the binary values of x_{ui} .

Algorithm 1 Alternating Least Squares training algorithm.

```

procedure ALS-TRAIN
  Initialize  $\mathbf{P}$ ,  $\mathbf{Q}$  at random
  repeat
    P step
      Fix  $\mathbf{Q}$  and optimize all  $\mathbf{p}_u$  in parallel using Equation 8
    Q step
      Fix  $\mathbf{P}$  and optimize all  $\mathbf{q}_i$  in parallel using Equation 9
  until convergence
end procedure

```

As pointed out by the authors, the products $\mathbf{Q}^\top \mathbf{C}^u \mathbf{Q}$ and $\mathbf{P}^\top \mathbf{C}^i \mathbf{P}$ require time $\mathcal{O}(k^2|U|)$ and $\mathcal{O}(k^2|I|)$ for each user and item, respectively, and represent a computational bottleneck during the training phase. However, these terms can be computed more efficiently noting that $\mathbf{Q}^\top \mathbf{C}^u \mathbf{Q} = \mathbf{Q}^\top \mathbf{Q} + \mathbf{Q}^\top (\mathbf{C}^u - \mathbf{I}) \mathbf{Q}$, where $\mathbf{Q}^\top \mathbf{Q}$ is independent of the user and thus can be precomputed, and $\mathbf{C}^u - \mathbf{I}$ only has non-zero entries in the diagonal for the $|I(u)|$ items with $k_{ui} > 0$, which is much smaller than $|I|$. Considering the computation of the matrix inverse, the total complexity of Eq. (8) for a single user is $\mathcal{O}(k^2|I(u)| + k^3)$. Likewise, the complexity for Eq. (9) is $\mathcal{O}(k^2|U(i)| + k^3)$.

The main advantage of ALS is that the optimal factors for each user in Eq. (8) can be computed in parallel once the item factors are fixed (P step). Symmetrically, once the user factors are obtained and fixed, the item factors in Eq. (9) can be found for each item in parallel (Q step). This observation leads to the alternating nature of ALS, respectively fixing one set of parameters and optimizing the other until convergence is reached or for a given number of iterations, as illustrated in Algorithm 1.

The ALS-based method by Hu et al. (2008) became the standard baseline for matrix factorization models with positive-only feedback, and has been extended and improved in subsequent works since it was first proposed. One notable paper by Pilászy et al. (2010) presents a new training procedure to boost the time complexity of the P step of each user to $\mathcal{O}(k^2 + k|I(u)|)$, and analogously the Q step. In order to achieve this significant improvement, the authors propose an approximate solution to the least-squares problem in each step. Rather than directly finding the k -dimensional solution as in Eqs. (8) and (9), which involves the costly computation of a matrix inverse, their approach iteratively solves k one-dimensional least squares problems, one for each latent dimension, much less expensive to solve. As reported in the paper, the loss of accuracy due to the approximate algorithm is small compared to the saved time for training. In subsequent work, Takács and Tikk (2012) extended ALS to a ranking-based MF approach that learns to predict the relative ordering of items instead of individual point-wise scores. More recently, Paquet and Koenigstein (2013) proposed a graph-based Bayesian model that is able to capture the meaning of missing values, distinguishing between a user disliking an item or being unaware of it.

3.3 Bayesian personalized ranking criterion

Matrix Factorization models can be easily trained to reduce the prediction error via gradient descent methods, alternating least-squares (ALS), and MCMC. If the problem

is formulated as a top-N recommendation task, they can be trained using a learning to rank approach like the Bayesian Personalized Ranking Criterion (BPR) from Rendle et al. (2009). Given Θ as the model parameters, and \succ_u the unknown preference structure we want to learn for user u , the optimal personalized ranking using Bayesian formulation can be found optimizing the posterior probability:

$$p(\Theta | \succ_u) \propto p(\succ_u | \Theta) \cdot p(\Theta)$$

The BPR criterion is optimized using a stochastic gradient descent algorithm on a set D_S of triples (u, i, j) , with $i \in I^u$ and $j \notin I^u$, selected through a random sampling from a uniform distribution. From the former, the overall optimization criterion BPR-OPT can be derived defining the individual probability with a sigmoid function $\sigma(\cdot)$:

$$\begin{aligned} \text{BPR-OPT} &= \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_{\Theta} \|\Theta\|^2 \\ &= \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{ui} - \hat{x}_{uj}) - \lambda_{\Theta} \|\Theta\|^2 \end{aligned} \quad (10)$$

Given the sigmoid function, the update step is defined as:

$$\Theta \leftarrow \Theta + \eta \left(\frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_{\Theta} \right) \quad (11)$$

where η is the chosen learning rate, and the values \hat{x}_{uij} are set to the difference of the predicted values, between a positive element (which is present in a user's history), and a negative element (randomly extracted from the remaining items). In the classical BPR formulation, a Biased Matrix Factorization model (Koren and Bell 2015) is adopted in which the prediction \hat{x}_{ui} corresponds to:

$$\hat{x}_{ui} = b_u + b_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle \quad (12)$$

where b_u and b_i are, respectively, the learned user and item biases. The difference of the predicted values, \hat{x}_{uij} , is thus:

$$\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj} = b_i - b_j + \langle \mathbf{p}_u, \mathbf{q}_i \rangle - \langle \mathbf{p}_u, \mathbf{q}_j \rangle \quad (13)$$

By computing the partial derivatives, the updates of the parameters for the model are:

$$b_u \leftarrow b_u + \eta \left(\frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} - \lambda b_u \right) \quad (14)$$

$$b_i \leftarrow b_i + \eta \left(-\frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} - \lambda b_i \right) \quad (15)$$

$$\mathbf{q}_i \leftarrow \mathbf{q}_i + \eta \left(\mathbf{p}_u \frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} - \lambda \mathbf{q}_i \right) \quad (16)$$

$$\mathbf{q}_j \leftarrow \mathbf{q}_j + \eta \left(-\mathbf{p}_u \frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} - \lambda \mathbf{q}_j \right) \quad (17)$$

$$\mathbf{p}_u \leftarrow \mathbf{p}_u + \eta \left[(\mathbf{q}_i - \mathbf{q}_j) \frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} - \lambda \mathbf{p}_u \right] \quad (18)$$

Using Eqs. (14–18) the model parameters can be iteratively updated to maximize the BPR–OPT criterion. The algorithm updates sequentially the model using each sampled triple and continues the training until it reaches the convergence (usually the provided number of iterations).

3.4 FISM

Matrix factorization has also been exploited to alleviate the sparsity problem in generating the item similarity matrix. In FISM, Kabbur et al. (2013) proposed to learn the item similarity matrix as the product of two low dimensional latent factor matrices. Given two items i and j , the similarity $sim(i, j)$ between them is computed as the dot product between the corresponding factors from \mathbf{P} and \mathbf{Q} , i.e., $sim(i, j) = \langle \mathbf{p}_j, \mathbf{q}_i \rangle$. The rating for a given user u on item i is estimated as

$$\hat{x}_{ui} = b_u + b_i + (|I| - 1)^{-\alpha} \sum_{j \in I_u \setminus \{i\}} \langle \mathbf{p}_j, \mathbf{q}_i \rangle \quad (19)$$

In order to solve the top-N recommendation problem, the authors used a ranking loss inspired by Bayesian Personalized Ranking (BPR) (Rendle et al. 2009), which optimizes the area under the curve (AUC). The consequent model, FISM_{AUC}, is learned solving the following minimization problem:

$$\begin{aligned} \underset{\mathbf{P}, \mathbf{Q}}{\text{minimize}} & \frac{1}{2} \sum_{(u,i,j) \in D_s} \|(x_{ui} - x_{uj}) - (\hat{x}_{ui} - \hat{x}_{uj})\|_k^2 + \frac{\beta}{2} (\|\mathbf{P}\|_k^2 + \|\mathbf{Q}\|_k^2) \\ & + \frac{\gamma}{2} (\|b_i\|_2^2) \end{aligned} \quad (20)$$

where the vectors b_i correspond to the vector of item biases. The regularization terms are used to prevent overfitting, and β and γ are the regularization weights for latent factor matrices, and item bias vector respectively.

4 Proposed matrix factorization models for cross-domain recommendation

In contrast to previous works that rely on graph-based methods for exploiting semantic metadata, the approach proposed in this paper first computes inter-domain content-

based similarities between the items, and then exploits these similarities to regularize the joint learning of matrix factorizations in the source and target domains. In particular, we present three alternative hybrid models that make different assumptions about the relationships between source and target domain item latent factors, simultaneously exploiting user preferences and item metadata.

Moreover, in the paper we detail our adaptations of the fast alternating least squares training algorithm for matrix factorization proposed by Pilászy et al. (2010), in order to deal with the increased complexity of our models, which not only learn the auxiliary source domain user preferences, but also the item metadata used to bridge the domains.

Items from different domains tend to have very diverse attributes that are not straightforward related. For instance, a book may be characterized by its *author* or by its *book genres*, and a movie can be described using its *cast*, *director* or *movie genres*. In fact, content-based features are often different between domains, and even when they refer to related concepts, such as *book genres* and *movie genres*, the features may not be directly aligned, e.g., *funny movies* vs. *comedy books*.

In order to overcome the heterogeneity of features of items from different domains, we propose to exploit Linked Data for linking entities from multiple and diverse domains. Specifically, we map the items in our datasets to entities in DBpedia (Lehmann et al. 2015), a multi-domain repository that provides a semantic-based, structured representation of knowledge in Wikipedia.⁹ In Sect. 5.1 we shall describe the process of mapping items to semantic entities from DBpedia. Once the items are mapped to their corresponding entities, we use the DBpedia graph to compute semantic similarities between such entities, mining both the attributes and the structure of the graph with semantic relations. More specifically, we exploit the information in DBpedia to compute a semantic similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{J}_S| \times |\mathcal{J}_T|}$ between the source domain items \mathcal{J}_S and the target domain items \mathcal{J}_T :

$$s_{ij} = \text{sim}(i, j), \quad i \in \mathcal{J}_S, j \in \mathcal{J}_T \quad (21)$$

In Sect. 5.4 we shall report recommendation performance results by using several semantic similarity metrics from the state of the art.

The computed inter-domain item similarities are then used to *link* the domains for cross-domain recommendation. In the cold-start, when a user has rated a few (if any) items in the target domain, a recommender system could suggest the user items in the target domain that are semantically similar to those the user liked in the source domain. Hence, the system could be effective only if there is an overlap of users between the domains. Moreover, even cold start users in the target domain should have some preferences in the source domain.

In the next subsections we present our three recommendation models based on the exploitation of semantic similarities to regularize item factors in MF, so that similar items from different domains tend to have similar parameters. In this way, even if the user's preferences in the target domain are unknown, a recommender system could suggest the user with target items that are most similar to those she preferred in the source.

⁹ Wikipedia online encyclopedia, <https://www.wikipedia.org>.

4.1 Regularization through similarity prediction

The first semantic-based matrix factorization cross-domain model we propose is based on the assumption that latent vectors of related items should explain the items semantic similarities, in addition to the users' preferences. That is, we not only seek to predict the preferences $r_{ui} \approx \langle \mathbf{p}_u, \mathbf{q}_i \rangle$, but also the inter-domain similarities $s_{ij} \approx \langle \mathbf{q}_i, \mathbf{q}_j \rangle$, where $i \in \mathcal{J}_S$ and $j \in \mathcal{J}_T$.

Hence, our model jointly factorizes the rating and inter-domain item similarity matrices that link the source and target domains. Let $\mathcal{U} = \mathcal{U}_S \cup \mathcal{U}_T$ be the set of all users, which we assume overlaps between the domains, and let $\mathcal{J} = \mathcal{J}_S \cup \mathcal{J}_T$ be the set of all items, which we assume do not overlap. Our model learns a latent vector $\mathbf{p}_u \in \mathbb{R}^k$ for each user $u \in \mathcal{U}$, but separately models source and target domain items \mathbf{q}_i and \mathbf{q}_j , with $i \in \mathcal{J}_S$ and $j \in \mathcal{J}_T$, as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{P}, \mathbf{Q}_S, \mathbf{Q}_T) = & \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{J}} c_{ua} (r_{ua} - \langle \mathbf{p}_u, \mathbf{q}_a \rangle)^2 \\ & + \lambda_C \sum_{i \in \mathcal{J}_S} \sum_{j \in \mathcal{J}_T} (s_{ij} - \langle \mathbf{q}_i, \mathbf{q}_j \rangle)^2 + \lambda (\|\mathbf{P}\|^2 + \|\mathbf{Q}_S\|^2 + \|\mathbf{Q}_T\|^2) \end{aligned} \quad (22)$$

where \mathbf{Q}_S and \mathbf{Q}_T are matrices containing the item latent vectors as rows from the source and target domains, respectively. We note that the summation in the first term iterates over all items $a \in \mathcal{J}$ from both domains, as we want to factorize the source and target user–item preference matrices simultaneously. The cross-domain regularization parameter $\lambda_C > 0$ controls the contribution of the inter-domain semantic similarities; large values of the parameter will force items to have too similar latent vectors, whereas low values will result in limited transfer of knowledge between domains.

As in standard matrix factorization, we train our model using Alternating Least Squares. First, we fix \mathbf{Q}_S and \mathbf{Q}_T , and solve analytically for each \mathbf{p}_u by setting the gradient to zero. Since the user factors do not appear in the additional cross-domain regularization term, we obtain the same solution as for the baseline MF model (see Eq. 8):

$$\mathbf{p}_u = \left(\mathbf{Q}^\top \mathbf{C}^u \mathbf{Q} + \lambda \mathbf{I} \right)^{-1} \mathbf{Q}^\top \mathbf{C}^u \mathbf{r}_u \quad (23)$$

In order to simplify the notation, we have defined the matrix \mathbf{Q} as the row-wise concatenation of \mathbf{Q}_S and \mathbf{Q}_T . The matrix \mathbf{C}^u is a diagonal matrix with the confidence values c_{ua} for all $a \in \mathcal{J}$, and the vector \mathbf{r}_u contains the preferences of user u , again for all items $a \in \mathcal{J}$.

Next, we fix the user factors \mathbf{P} and the target domain item factors \mathbf{Q}_T , and compute the optimal values for the source domain item factors. Again, by setting the corresponding gradient to zero and solving analytically we obtain:

$$\mathbf{q}_i = \left(\mathbf{P}^\top \mathbf{C}^i \mathbf{P} + \lambda_C \mathbf{Q}_T^\top \mathbf{Q}_T + \lambda \mathbf{I} \right)^{-1} \left(\mathbf{P}^\top \mathbf{C}^i \mathbf{r}_i + \lambda_C \mathbf{Q}_T^\top \mathbf{s}_i \right) \quad (24)$$

As previously, the vector \mathbf{r}_i contains the preferences assigned to item i , and \mathbf{s}_i is the i th row of the inter-domain semantic similarity matrix \mathbf{S} . Finally, we proceed as before

fixing \mathbf{P} and \mathbf{Q}_S to compute the optimal solution for the target domain item latent vectors:

$$\mathbf{q}_j = \left(\mathbf{P}^\top \mathbf{C}^j \mathbf{P} + \lambda_C \mathbf{Q}_S^\top \mathbf{Q}_S + \lambda \mathbf{I} \right)^{-1} \left(\mathbf{P}^\top \mathbf{C}^j \mathbf{r}_j + \lambda_C \mathbf{Q}_S^\top \mathbf{s}_j \right) \quad (25)$$

The computation of the optimal factors can be parallelized within each step, but the larger number of items to consider and the extra step required for the source domain greatly increase the training time with respect to the MF baseline. In order to address this issue, we adapt the fast RR1 training algorithm for ALS proposed by Pilászy et al. (2010). Since the computation of the user factors is the same as in the original MF model, the procedure remains the same for the P-step. For the source domain Q-step, by inspecting Eqs. (22) and (24), we note that the additional terms that arise from the inter-domain similarities can be treated just like user preferences as follows. For each source item i :

1. Generate examples for each rating r_{ui} as for baseline MF [see Pilászy et al. (2010)]
2. For each target item $j \in \mathcal{J}_T$:
 - Generate an input example $\mathbf{x}_j := \mathbf{q}_j$.
 - Use the similarity as the dependent variable, $y_j := s_{ij}$.
 - Use a constant confidence value $c_j := \lambda_C$.
 - The parameter to optimize is $\mathbf{w} := \mathbf{q}_j$.

The above procedure will produce the similarity terms of Eq. 24, which can be defined by means of the confidence matrix $\tilde{\mathbf{C}}^i = \lambda_C \mathbf{I}$. The procedure for the target domain Q-step is completely analogous.

4.2 Regularization based on item neighborhoods

Our second semantic-based matrix factorization cross-domain model exploits the item semantic similarities in a different fashion. Instead of forcing pairwise item interactions to reproduce the observed similarity values, the approach we present here leverages \mathbf{S} to regularize the item latent vectors, so that feature vectors of similar items are pushed together in the latent space. Intuitively, items that are semantically similar should also have similar latent parameters.

As previously, let $\mathcal{U} = \mathcal{U}_S \cup \mathcal{U}_T$ and $\mathcal{J} = \mathcal{J}_S \cup \mathcal{J}_T$ be the sets of all users and items, respectively. Our approach jointly factorizes the source and target domain rating matrices, and regularizes similar item factors proportionally to the items similarity. However, instead of considering all the potentially similar source domain items, we limit the regularization of a target domain item $j \in \mathcal{J}_T$ to its neighborhood, i.e., to the set $N(j) \subseteq \mathcal{J}_S$ of the top- n most similar source domain items:

$$\begin{aligned} \mathcal{L}(\mathbf{P}, \mathbf{Q}_S, \mathbf{Q}_T) = & \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{J}} c_{ua} (r_{ua} - \langle \mathbf{p}_u, \mathbf{q}_a \rangle)^2 \\ & + \lambda_C \sum_{j \in \mathcal{J}_T} \sum_{i \in N(j)} s_{ij} \|\mathbf{q}_j - \mathbf{q}_i\|^2 + \lambda \left(\sum_{u \in \mathcal{U}} \|\mathbf{p}_u\|^2 + \sum_{a \in \mathcal{J}} \|\mathbf{q}_a\|^2 \right) \end{aligned} \quad (26)$$

We note that items with greater similarity values are more heavily regularized, whereas items with values of $s_{ij} \approx 0$ in their neighborhoods are barely affected. However, it may still be convenient to regularize such items so that they benefit from cross-domain information, and thus may be eligible for recommendation to cold start users. Therefore, we also experiment normalizing the similarity scores in the item neighborhoods so that $\sum_{i \in N(j)} s_{ij} = 1$. In this way all target items are equally regularized, but each is affected by its source domain neighbors proportionally to their similarity scores.

By assigning latent vectors to target domain items close to those of similar source domain items, our model is able to generate recommendations in cold start settings. Specifically, let \mathbf{q}_j be the latent vector learned for target item $j \in \mathcal{J}_T$, and let \mathbf{q}_i be the latent vector of source item $i \in \mathcal{J}_S$, which we assume is semantically similar to j . Our model will regularize both factors so that their distance $\|\mathbf{q}_j - \mathbf{q}_i\|$ is small, or equivalently, $\mathbf{q}_j \approx \mathbf{q}_i$. Consider now a cold start user u who only provided preferences in the source domain, so that her corresponding latent vector \mathbf{p}_u is therefore only adjusted using source domain preferences. In standard MF, it is not guaranteed that \mathbf{p}_u will extrapolate to the target domain, and will provide an accurate prediction for \mathbf{q}_j . In contrast, our model ensures that $\langle \mathbf{p}_u, \mathbf{q}_j \rangle \approx \langle \mathbf{p}_u, \mathbf{q}_i \rangle$, i.e., target domain items yield relevance prediction scores close to that of similar source domain items. Hence, u will be recommended with a target domain item j if the user liked the source domain item i , or if i would be recommended to u in the source domain.

Once more, we train our neighborhood-based matrix factorization model using Alternating Least Squares. As in the previous model, the user factors are not affected by the extra regularization, and can be computed again using Eq. (23), leaving the P-step unchanged. For the target domain item factors \mathbf{q}_j we proceed as usual, fixing the user and source item factors, and finding the values such that $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_j} = 0$, which yields the solution:

$$\mathbf{q}_j = \left[\mathbf{P}^\top \mathbf{C}^j \mathbf{P} + \left(\lambda + \lambda_C \sum_{i \in N(j)} s_{ij} \right) \mathbf{I} \right]^{-1} \left(\mathbf{P}^\top \mathbf{C}^j \mathbf{r}_j + \lambda_C \sum_{i \in N(j)} s_{ij} \mathbf{q}_i \right) \quad (27)$$

Repeating the same procedure for the source item factors \mathbf{q}_i we obtain:

$$\mathbf{q}_i = \left[\mathbf{P}^\top \mathbf{C}^i \mathbf{P} + \left(\lambda + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij} \right) \mathbf{I} \right]^{-1} \left(\mathbf{P}^\top \mathbf{C}^i \mathbf{r}_i + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij} \mathbf{q}_j \right) \quad (28)$$

where $N^{-1}(i)$ is the *inverse neighborhood* of item i , i.e., the set of target domain items that have i among their neighbors: $N^{-1}(i) = \{j \in \mathcal{J}_T | i \in N(j)\}$.

Unlike the model presented in the previous section, we cannot apply RR1 directly by treating the new similarity terms as additional user preferences. Instead, we derive again the update rules for each component of the source and target domain item parameters. As mentioned before, user parameters remain unchanged. Let $j \in \mathcal{J}_T$ be a target item, and consider the optimization of the α th component $q_{j\alpha}$ of its corresponding

latent vector \mathbf{q}_j . We can rewrite the loss in Eq. (26) as a function only of $q_{j\alpha}$ as follows:

$$\begin{aligned} \mathcal{L}_\alpha(q_{j\alpha}) &= \sum_{u \in \mathcal{U}} c_{uj} (e_{uj} - p_{u\alpha} q_{j\alpha})^2 + \lambda q_{j\alpha}^2 \\ &\quad + \lambda_C \sum_{i \in N(j)} s_{ij} (q_{j\alpha} - q_{i\alpha})^2 + \text{constant} \end{aligned} \quad (29)$$

where $e_{uj} \triangleq r_{uj} - \sum_{\beta \neq \alpha} p_{u\beta} q_{j\beta}$, and the constant includes terms that do not depend on $q_{j\alpha}$. If we set the derivative $\frac{d\mathcal{L}_\alpha}{dq_{j\alpha}} = 0$, we obtain:

$$q_{j\alpha} = \frac{\sum_{u \in \mathcal{U}} c_{uj} e_{uj} p_{u\alpha} + \lambda_C \sum_{i \in N(j)} s_{ij} q_{i\alpha}}{\sum_{u \in \mathcal{U}} c_{uj} p_{u\alpha}^2 + \lambda + \lambda_C \sum_{i \in N(j)} s_{ij}} \quad (30)$$

Using the optimizations described in Pilászy et al. (2010), the computational cost of the above formula for all items is $\mathcal{O}(k^2|\mathcal{U}| + k|\mathcal{R}| + n|\mathcal{J}_T|)$, since all the neighborhoods are formed using the top n most similar items, $|N(j)| \leq n$. Applying the same procedure to the source domain item factor \mathbf{q}_i we obtain:

$$q_{i\alpha} = \frac{\sum_{u \in \mathcal{U}} c_{ui} e_{ui} p_{u\alpha} + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij} q_{j\alpha}}{\sum_{u \in \mathcal{U}} c_{ui} p_{u\alpha}^2 + \lambda + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij}} \quad (31)$$

The main difference with respect to Eq. (30) is that the sets $N^{-1}(i)$ are not bounded, as a source item can potentially be the neighbor of an arbitrary number of target items, so that $|N^{-1}(i)| \leq |\mathcal{J}_T|$, resulting in a theoretical worst-case cost of $\mathcal{O}(k^2|\mathcal{U}| + k|\mathcal{R}| + |\mathcal{J}_S||\mathcal{J}_T|)$. We observe, however, that in practice most of the source items appear only in a few neighborhoods and that the algorithm is still very efficient.

4.3 Regularization based on item centroids

When neighbor source domain items are mutually diverse, the neighborhood-based model presented in the previous section may struggle to regularize a target domain item that has to be simultaneously close to all its neighbors. The model we propose in this section works like the neighborhood-based model, but, instead of using the neighbor source domain items *individually* in the regularization, it uses their *centroid* (average) latent vector:

$$\begin{aligned} \mathcal{L}(\mathbf{P}, \mathbf{Q}_S, \mathbf{Q}_T) &= \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{J}} c_{ua} (r_{ua} - \langle \mathbf{p}_u, \mathbf{q}_a \rangle)^2 \\ &\quad + \lambda_C \sum_{j \in \mathcal{J}_T} \left\| \mathbf{q}_j - \sum_{i \in N(j)} s_{ij} \mathbf{q}_i \right\|^2 + \lambda \left(\sum_{u \in \mathcal{U}} \|\mathbf{p}_u\|^2 + \sum_{a \in \mathcal{J}} \|\mathbf{q}_a\|^2 \right) \end{aligned} \quad (32)$$

The same considerations regarding the neighborhood $N(j)$ and the normalization of the similarity scores also apply to this model. However, the effect on the item relevance predictions for cold start users is different. Let \mathbf{q}_j be an item in the target domain, and let $N(j)$ be its neighborhood of most similar source domain items. The regularization scheme in our centroid-based approach aims to minimize the distance $\left\| \mathbf{q}_j - \sum_{i \in N(j)} s_{ij} \mathbf{q}_i \right\|$, so that the latent vector of item j is close, *on average*, to those of the source items in $N(j)$, i.e., $\mathbf{q}_j \approx \sum_{i \in N(j)} s_{ij} \mathbf{q}_i$. Let u be a cold start user in the target domain that has some preferences in the source domain. Again, her feature vector \mathbf{p}_u is only learned using the user’s source preferences, and may not be reliable for computing relevance predictions for target domain items in standard MF. Our model, however, ensures that

$$\langle \mathbf{p}_u, \mathbf{q}_j \rangle \approx \left\langle \mathbf{p}_u, \sum_{i \in N(j)} s_{ij} \mathbf{q}_i \right\rangle = \sum_{i \in N(j)} s_{ij} \langle \mathbf{p}_u, \mathbf{q}_i \rangle$$

That is, the predicted relevance score is roughly the average of the relevance scores for the neighbor source domain items, weighted by their corresponding semantic similarity.

As in the previous models, the user parameters are not affected by the item regularization terms, and can be computed in the standard fashion using Eq. (23). For the target domain item factors $\mathbf{q}_j, j \in \mathcal{J}_T$, we set the gradient of Eq. (32) to zero to obtain:

$$\mathbf{q}_j = \left[\mathbf{P}^\top \mathbf{C}^j \mathbf{P} + (\lambda + \lambda_C) \mathbf{I} \right]^{-1} \left(\mathbf{P}^\top \mathbf{C}^j \mathbf{r}_j + \lambda_C \sum_{i \in N(j)} s_{ij} \mathbf{q}_i \right) \tag{33}$$

Comparing the above to Eq. (27) we observe that both are equivalent when $\sum_{i \in N(j)} s_{ij} = 1$, i.e., normalizing the similarity values has the same effect of the centroid-based regularization on the target domain item factors. The solution for source item factors \mathbf{q}_i , in contrast, has a different form:

$$\begin{aligned} \mathbf{q}_i = & \left[\mathbf{P}^\top \mathbf{C}^i \mathbf{P} + \left(\lambda + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij}^2 \right) \mathbf{I} \right]^{-1} \\ & \times \left(\mathbf{P}^\top \mathbf{C}^i \mathbf{r}_i + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij} (\mathbf{q}_j - \mathbf{z}_{j \setminus i}) \right) \end{aligned} \tag{34}$$

where we have defined $\mathbf{z}_{j \setminus i} = \sum_{l \in N^{-1}(i), l \neq j} s_{lj} \mathbf{q}_l$ to simplify the notation. We note that, differently to the previous models, the computation of the source domain latent vectors cannot be parallelized, as the value of $\mathbf{q}_i, i \in \mathcal{J}_S$ depends on the values of other $\mathbf{q}_l, l \in \mathcal{J}_S$ through the parameter $\mathbf{z}_{j \setminus i}$. As a result, the training process can be slow when the set of source domain items is large. In our experiments, however, we

observed that the time penalty of computing the source factors sequentially is usually compensated by the faster RR1 algorithm, although we do not provide any quantitative analysis as it falls out of the scope of this work.

In order to apply RR1 to our centroid-based approach, we derive again the solutions for each α th coordinate separately. Once more, the solution for the user factors remains the same as it is not affected by the regularization terms. For the target domain item factors \mathbf{q}_j , we consider the loss in Eq. (32) as a function only of the α th component $q_{j\alpha}$:

$$\begin{aligned} \mathcal{L}_\alpha(q_{j\alpha}) = & \sum_{u \in \mathcal{U}} c_{uj} (e_{uj} - p_{u\alpha} q_{j\alpha})^2 + \lambda q_{j\alpha}^2 \\ & + \lambda_C \left(q_{j\alpha} - \sum_{i \in N(j)} s_{ij} q_{i\alpha} \right)^2 + \text{constant} \end{aligned} \quad (35)$$

As previously, the constant includes terms that do not depend on $q_{j\alpha}$, and e_{uj} is defined as in Eq. (29). Setting the derivative $\frac{d\mathcal{L}_\alpha}{dq_{j\alpha}} = 0$ yields:

$$q_{j\alpha} = \frac{\sum_{u \in \mathcal{U}} c_{uj} e_{uj} p_{u\alpha} + \lambda_C \sum_{i \in N(j)} s_{ij} q_{i\alpha}}{\sum_{u \in \mathcal{U}} c_{uj} p_{u\alpha}^2 + \lambda + \lambda_C} \quad (36)$$

We note, once again, the similar form of the above solution with respect to the previous model in Eq. (30). If we apply the same procedure to the source domain item factors, we obtain:

$$q_{i\alpha} = \frac{\sum_{u \in \mathcal{U}} c_{ui} e_{ui} p_{u\alpha} + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij} (q_{j\alpha} - \mathbf{z}_{(j \setminus i)\alpha})}{\sum_{u \in \mathcal{U}} c_{ui} p_{u\alpha}^2 + \lambda + \lambda_C \sum_{j \in N^{-1}(i)} s_{ij}^2} \quad (37)$$

The computational complexity for the target domain factors is equivalent to the model from the previous section, whereas for the source domain factors it is $\mathcal{O}(k^2|\mathcal{U}| + k|\mathcal{R}| + n|\mathcal{J}_S||\mathcal{J}_T|)$ in the worst case, which is similar to the neighborhood-based model since the size of the neighborhoods n is in general small.

5 Experiments

In a first experiment, we compared several state-of-the-art semantic similarity metrics for content-based recommendation, aiming to understand which is more suitable for later injecting in our cross-domain MF models, and achieved the best results using the link-based approach by Milne and Witten (2008). Second, we evaluated the ranking precision and diversity of the recommendations computed by the proposed models. We show that, depending on the involved source and target domains, our models generate more accurate suggestions than the baselines in severe cold start situations. Moreover, the proposed approaches provide a better trade-off between accuracy and diversity, which are in general difficult to balance.

5.1 Dataset

Our dataset¹⁰ initially consisted of a large set of *likes* assigned by users to items on Facebook. Using the Facebook Graph API, a user's *like* is retrieved in the form of a 4-tuple with the following information: the identifier, name and category of the liked item, and the timestamp of the like creation, e.g., {id: "35481394342", name: "The Godfather", category: "Movie", created_time: "2015-05-14T12:35:08+0000"}. The name of an item is given by the user who created the Facebook page of such item. In this context, distinct names may exist for a particular item, e.g., *The Godfather*, *The Godfather: The Movie*, *The Godfather - Film series*, etc. Users thus may express likes for different Facebook pages which actually refer to the same item. Aiming to unify and consolidate the items extracted from Facebook likes, we developed a method that automatically maps the items names with the unique URIs of the corresponding DBpedia entities, e.g., http://dbpedia.org/resource/The_Godfather for the identified names of *The Godfather* movie.

Linking items to DBpedia entities Given a particular item, we first identified DBpedia entities that are labeled with the name of the item. For such purpose, we launched a SPARQL query targeted on the subjects of triples that have `rdfs:label`¹¹ as property and the item title as object. The next query is an example for *The Matrix 2* title:

```
SELECT DISTINCT ?item WHERE {
  {
    ?item rdf:type dbo:Film .
    ?item rdfs:label ?name .
    FILTER regex(?name, "the.*matrix.*2", "i") .
  }
  UNION
  {
    ?item rdf:type dbo:Film .
    ?tmp dbo:wikiPageRedirects ?item .
    ?tmp rdfs:label ?name .
    FILTER regex(?name, "the.*matrix.*2", "i") .
  }
}
```

To resolve ambiguities in those names that correspond to multiple items belonging to different domains, we specify the type of the item we wanted to retrieve in each case. Specifically, the previous query includes a triple clause with `rdf:type`¹² (or `dbo:type`¹³) as property. Hence, in the given example, the subject *The Matrix 2* refers to the “movie” type, which is associated to the `dbo:Film` class in DBpedia.

¹⁰ Available at <http://ir.ii.uam.es/metadata>.

¹¹ Namespace for rdfs, <http://www.w3.org/2000/01/rdf-schema>.

¹² Namespace for rdf, <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

¹³ Namespace for dbo, <http://dbpedia.org/ontology>.

Table 1 Considered item types and their DBpedia and YAGO classes for the three domains of the dataset

Item type	DBpedia/YAGO classes
Books	
Book	dbo:Book, yago:Book102870092, yago:Book102870526
Genre	yago:LiteraryGenres
Writer	dbo:Writer, yago:Writer110794014
Fictional character	dbo:FictionalCharacter, yago:FictionalCharacter109587565
Movies	
Movie	dbo:Film, yago:Movie106613686
Genre	dbo:MovieGenre, yago:FilmGenres
Director	yago:FilmDirector110088200, yago:Director110014939
Actor	dbo:Actor, yago:Actor109765278
Fictional character	dbo:FictionalCharacter, yago:FictionalCharacter109587565
Music	
Composition	dbo:Song, dbo:MusicalWork, dbo:Single, dbo:ClassicalMusicComposition, dbo:Opera
Genre	dbo:MusicGenre, yago:MusicGenres, yago:MusicGenre107071942
Album	dbo:Album, yago:Album106591815
Musician	dbo:MusicalArtist, yago:Musician110339966, yago:Musician110340312, yago:Composer109947232
Band	dbo:Band, yago:MusicalOrganization108246613

The item types were set from the item categories provided in Facebook, and their associated DBpedia and YAGO¹⁴ classes¹⁵ were identified by manual inspection of the `rdf:type` values of several entities. Table 1 shows the list of item types and DBpedia/YAGO classes we considered for the three domains of our dataset.

Moreover, running the previous query template we observed that a number of items were not linked to DBpedia entities because the labels corresponded to Wikipedia *redirection* webpages. In these cases, to reach the appropriate entities the query makes use of the `dbo:wikiPageRedirects` property. The result of the previous query for *The Matrix 2* is http://dbpedia.org/resource/The_Matrix_Reloaded, which actually is the DBpedia entity of the second movie in *The Matrix* saga. Here, it is important to note that thanks to the Wikipedia page redirect component we were able to link items whose names do not have a direct syntactic match with the label of its DBpedia entity, but with the label of a redirected entity, e.g., the *Matrix 2* title matches the *The Matrix Reloaded* entity.

Final semantically annotated dataset For every linked entity, we finally accessed DBpedia to retrieve the metadata that afterward will be used as input for the recommendation models. In this case, we launched a SPARQL query asking for all the

¹⁴ The YAGO knowledge base, <http://www.mpi-inf.mpg.de/yago-naga/yago>.

¹⁵ Namespace for yago, <http://dbpedia.org/class/yago>.

Table 2 DBpedia properties considered as item metadata; *item* can be book, movie and composition, musician and band

Relation	DBpedia properties
Item–genre	dct:subject, dbo:genre
Book–genre	dbo:literaryGenre
Music genre–music genre	dbo:musicSubgenre, dbo:musicFusionGenre, dbo:movement, dbo:derivative, dbo:stylisticOrigin
Item–author	dbo:author, dbo:creator
Book–writer	dbo:writer
Movie–actor, character, director	dbo:starring, dbo:cinematography, dbo:director
Composition–musician	dbo:artist, dbo:composer, dbo:musicComposer, dbo:musicalArtist, dbo:associatedMusicalArtist
Music item–album	dbo:album
Band–musician	dbo:bandMember, dbo:formerBandMember, dbo:musicalBand, dbo:associatedBand
Item–item, character	dbo:series
Item–character	dbo:portrayer
Item–item	dbo:basedOn, dbo:previousWork, dbo:subsequentWork, dbo:notableWork

properties and objects of the triples that have the target entity as subject. Following the example given before, such a query would be:

```
SELECT ?p ?o WHERE {
  dbr:The_Matrix_Reloaded ?p ?o .
}
```

This query returns all the DBpedia property-value pairs of the `dbr:The_Matrix_Reloaded`¹⁶ entity. However, since our ultimate goal is item recommendation, we should only exploit metadata that may be relevant to relate common preferences of different users. Thus, we filtered the query results by considering certain properties in each domain. Specifically, Table 2 shows the list of DBpedia properties selected for each of the three domains of our dataset. Hence, for example, for the movie items, we would have as metadata the movies genres, directors, and actors, among others.

The items and relations shown in the table thus represent a *semantic network* that is automatically obtained from DBpedia for each particular domain. Table 3 shows statistics of the dataset for the three domains of interest, namely books, movies, and music. Additionally, users may express preferences in more than one domain. Table 4 shows the number of users shared between each pair of domains.

Semantically enriched item profiles Fixing *books*, *movies*, *musicians* and *bands* as the target items to be recommended, we can distinguish the following three types of item metadata obtained:

¹⁶ Namespace for dbr, <http://dbpedia.org/resource>.

Table 3 Statistics of the extracted dataset enriched with metadata

	Books	Movies	Music
Users	1876	26,943	49,369
Items	3557	3901	5748
Likes	42,869	876,501	2,084,462
Sparsity (%)	99.4	99.2	99.3
Avg. items/user	22.85	32.53	42.22
Avg. users/item	12.05	224.69	362.64

Table 4 User overlap between domains. To the right of each target, the ratio of shared users relative to the source domain

Source	Target					
	Books	%	Movies	%	Music	%
Books	1876	100.0	1495	79.7	1519	81.0
Movies	1495	5.5	26,943	100.0	21,720	80.6
Music	1519	3.1	21,720	44.0	49,369	100.0

- *attributes*, which correspond to item-attribute entities associated to the considered item types of Table 2, and are distinct to the entities of target items, e.g., the genre(s), director(s) and actors of a particular movie.
- *related items*, which correspond to the item-item properties in Table 2 that derive related entities, e.g., the novel a movie is based on (`dbo:basedOn` property), the prequel/sequel of a movie (`dbo:previousWork`/`dbo:subsequentWork` properties), or the musicians belonging to a band (`dbo:bandMember` property).
- *extended attributes*, which correspond to attribute-attribute properties that generate extended item attributes, originally not appearing as metadata, e.g., the subgenres of a particular music genre (`dbo:musicSubgenre` property).

The above three types of item metadata constitute the semantically enriched item profiles that we propose to use in our recommendation models. We note that they differ from the commonly used content-based item profiles composed of plain attributes. We also note that in the conducted experiments, the results achieved by exploiting the enriched profiles were better than those achieved by only using item attributes.

5.2 Evaluation methodology and metrics

The evaluation of the proposed models was conducted utilizing a modified user-based five-fold cross-validation strategy, based on the methodology by Kluver and Konstan (2014) for cold-start evaluation. Our goal is to understand how the different approaches perform as the number of observed *likes* in the target domain increases. First, we divide the set of users into five subsets of roughly equal size. In each cross-validation stage, we keep all the data from four of the groups in the training set. Then, for each user u

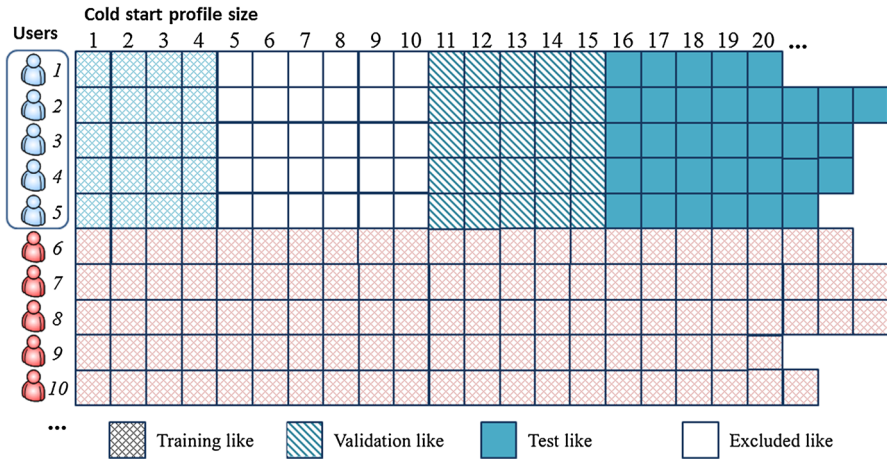


Fig. 1 Overview of the cold start evaluation setting in a given cross-validation fold. The box indicates the test users in the current fold, whose profiles are split into training, validation, and testing sets. Different cold start profile sizes are simulated by sequentially adding *likes* to their training sets—four in the figure

in the fifth group—the test users—we randomly split her *likes* into three subsets, as depicted in Fig. 1:

1. *Training data* initially filled with u 's *likes* and iteratively downsampled discarding one by one to simulate different cold start profile sizes,
2. *Validation data* containing the set of *likes* used for tuning hyperparameters, and
3. *Testing data* used to compute the performance metrics.

The above procedure was modified for the cross-domain scenario by extending the training set with the full set of *likes* from the auxiliary domain, in order to obtain the actual training data for the predictive models. For each cold start profile size, we built the recommendation models using the data in the final training set. Then, for each test user, we generated a ranked list of the top 10 suggested items from the set of target domain items in the training set that are not yet known to the user. The performance is estimated from the output of each model and the test set using rank-based metrics. We note that in our evaluation, any item ranked after position 10 by the model is considered not relevant when computing the metrics, as we are interested in the more realistic setting where the user only examines a limited subset of the recommendations.

Regarding the metrics, we used the Mean Reciprocal Rank (MRR) to evaluate the ranking accuracy of the recommendations, which computes the average reciprocal rank of the first relevant item in the recommendation list. We also examined other metrics such as Precision and MAP, but we found similar behavior to MRR so for brevity we omit them in our results. Binomial Diversity Framework (BinomDiv) (Vargas et al. 2014) was used to evaluate the individual diversity, namely the degree of diversity in the recommendation lists based on item genres extracted from DBpedia.

5.3 Evaluated methods

We compared the performance of our proposed methods against the following baseline algorithms:

- *POP* Non personalized baseline that always recommends the most popular items not yet liked by the user. Popularity is measured as the number of users in the dataset that liked the item.
- *UNN* User-based nearest neighbors with Jaccard similarity. The size of the neighborhood is tuned for each dataset using a validation set.
- *INN* Item-based nearest neighbors with Jaccard similarity and indefinite neighborhood size.
- *iMF* Matrix factorization method for positive-only feedback (Hu et al. 2008) trained using the fast ALS technique by Pilászy et al. (2010).
- *BPR* Bayesian personalized ranking from implicit feedback (Rendle et al. 2009). We used for our experiments the implementation available in LibRec (Guo et al. 2015).
- *FISM* Factored item similarity model by Kabbur et al. (2013). We used the implementation of the FISMauc variant optimized for the item ranking problem available in LibRec (Guo et al. 2015).
- *HeteRec* Graph-based recommender system proposed in Yu et al. (2014), based on a diffusion method of user preferences following different meta-paths.
- *SPRank* Originally proposed in Di Noia et al. (2016), it implements a hybrid approach to compute recommendations with LOD datasets. We used a publicly available implementation of SPRank.¹⁷

With the exception of POP (which only uses target domain data) and SPRank, we considered the application of all the baselines to both single- and cross-domain scenarios. We were not able to compute meaningful results for SPRank by using DBpedia properties shown in Table 2 due to the structure of the connections between domains in the underlying knowledge graph. All the paths calculated by SPRank to link items in different domains resulted in being not very relevant thus bringing to shallow performances of the algorithm. Moreover, given the datasets adopted for the experimental evaluation, we were not able to generate all the meta-paths needed to compute recommendations. We used machines with up to 3 TB of disk space but it was not sufficient.

Hereafter we use the prefix CD- to indicate that the algorithm is operating in cross-domain mode using the union of the rating matrices from the source and target domains. We did not consider for our evaluation the SemanticSVD++ method by Rowe (2014), as it is designed for rating prediction rather than item ranking. Moreover, preliminary tests showed that its performance was much lower than the other methods, and that its training time was about one order of magnitude larger.

We measured the performance of the three methods presented in this paper¹⁸ against the previous baselines:

- *SimMF* Our matrix factorization model regularized with similarity prediction described in Sect. 4.1.

¹⁷ <https://github.com/sisinflab/lodreclib>.

¹⁸ Code available at <https://github.com/nachoft/cross-metadata-mf>.

- *NeighborMF* Our proposed matrix factorization model with neighborhood-based regularization from Sect. 4.2.
- *CentroidMF* Our matrix factorization model from Sect. 4.3 that uses the neighbor's centroid to regularize the target domain item factors.

We tuned the hyperparameters of the considered recommendation models using a held-out validation set of likes, as we explain in the next section. For UNN, we only had to select the size of the user neighborhoods. For the matrix factorization models, in contrast, the number of hyperparameters is larger, namely, the dimensionality of the latent factor space k , the amount of regularization λ , and the confidence parameter for positive-only feedback α . Moreover, the models proposed in this paper also include the cross-domain regularization rate λ_C , which controls the contribution of the inter-domain item similarities. Finally, for NeighborMF and CentroidMF, we tuned the size n of the item neighborhoods $N(j)$, and the possibility to normalize the neighbors' similarities so that the sum to 1, as explained in Sect. 4.2.

The high number of parameters to tune rules out the possibility of performing a grid search for the best values. Hence, we used Bayesian Optimization techniques (Snoek et al. 2012) that train Machine Learning models to predict candidate values that are likely to maximize a given function while simultaneously reducing the uncertainty over unknown parameter values.

We tuned the parameters of the single-domain methods for each profile size only on the target domain. For simplicity, we used the same values for their cross-domain variants due to the combinatorial explosion of possible configurations. A quick inspection on books-movies and movies-music with profile size 3 did not reveal significant differences between the parameters on average. However, we acknowledge that our experiments could be improved by individually tuning each algorithm in each configuration, which we leave for future work. For UNN, the optimal number of neighbors was $n = 50$ for books, and $n = 100$ for movies and music. For iMF we obtained the optimal parameters $k = (10, 29, 21)$, $\lambda = (10^{-5}, 0.823, 1)$, and $\alpha = (6, 7, 10)$ for books, movies, and music, respectively. For BPR we used $\lambda = 0.01$ for regularization and $\eta = 0.01$ as learning rate. In the case of FISM, we used $\lambda = 0.001$ and $\eta = 10^{-5}$. The optimal values for our proposed cross-domain models are reported in Table 5.

5.4 Results

In this section we present the results of the conducted experiments to evaluate the proposed matrix factorization models. First, we analyze several semantic relatedness metrics to compute the inter-domain item similarities. Next, we report the ranking accuracy and diversity of the evaluated recommendation approaches, and study how the size and diversity of the source domain user profile impacts on the target recommendations.

Table 5 Optimal hyperparameters for SimMF, NeighborMF, and CentroidMF. The last column indicates whether the similarities in the neighborhood are normalized or not

Source	Method	k	λ	α	λ_C	n	Norm.
Books							
Movies	SimMF	112	0	1	10^{-8}		
	NeighborMF	134	1	1	9.125	49	✓
	CentroidMF	153	0.999	1	8.778	100	✓
Music	SimMF	10	1	16	10^{-8}		
	NeighborMF	10	0	18	10	100	✓
	CentroidMF	10	0	14	0.109	100	
Movies							
Books	SimMF	12	1	1	0.002		
	NeighborMF	12	1	1	10	81	✓
	CentroidMF	14	0.100	1	0.200	1	✓
Music	SimMF	35	0	1	1.6×10^{-6}		
	NeighborMF	51	1	1	10	100	
	CentroidMF	29	1	1	9.494	99	✓
Music							
Books	SimMF	10	1	1	0.039		
	NeighborMF	10	0.995	1	3.014	100	✓
	CentroidMF	10	0.724	1	1.673	14	
Movies	SimMF	11	0.571	4	0.641		
	NeighborMF	10	0.978	2	0.699	46	
	CentroidMF	10	0.562	2	10	3	✓

5.5 Inter-domain item semantic similarity

The goal of our first experiment is to analyze the performance of several semantic relatedness metrics to compute the inter-domain similarities that we later exploit in our matrix factorization models. We considered the following strategies:

- *TF-IDF* Semantically-enriched item profiles (see Sect. 5.1) are used to build TF-IDF vector profiles based on the metadata of each item. Given an item i , the enriched profile corresponds to a set of paths $\langle i, \rho, e \rangle$ where ρ is a predicate (or a sequence of predicates) in the *semantic network* \mathcal{SN} of a domain \mathcal{D} . The considered predicates are those depicted in Table 2, and described as *attributes*, *related items*, and *extended attributes*. We may build the set of all possible metadata for all items as $F = \{\langle \rho, e \rangle \mid \langle i, \rho, e \rangle \in \mathcal{SN} \text{ with } i \in I\}$. Each item can be then represented as a vector of weights $\omega_{\mathbf{i}} = [\omega_{(i,1)}, \dots, \omega_{(i,(\rho,e))}, \dots, \omega_{(i,|F|)}]$ where $\omega_{(i,(\rho,e))}$ is computed as the normalized TF-IDF value for $\langle \rho, e \rangle$:

$$\omega_{(i, \langle \rho, e \rangle)} = \underbrace{\frac{|\{\langle \rho, e \rangle \mid \langle i, \rho, e \rangle \in \mathcal{SN}\}|}{\sqrt{\sum_{\langle \rho, e \rangle \in F} |\{\langle \rho, e \rangle \mid \langle i, \rho, e \rangle \in \mathcal{SN}\}|^2}}}_{TF} \times \log \frac{|I|}{\underbrace{|\{j \mid \langle j, \rho, e \rangle \in \mathcal{SN} \text{ and } j \in I\}|}_{IDF}}$$

The similarity score between a source domain item and a target domain item is computed as the cosine of their corresponding TF-IDF vectors.

- *ESA* The Explicit Semantic Analysis technique proposed by Gabrilovich and Markovitch (2007). Instead of using the semantic metadata, we map each item to its corresponding Wikipedia article. Then, based on the text of the article, ESA extracts a set of other related Wikipedia articles, which represent semantic concepts, and builds a TF-IDF profile from the extracted concepts. Finally, the similarity score between two items is computed as the cosine of their corresponding concept-based vectors.
- *M&W* The approach proposed by Milne and Witten (2008) computes the semantic relatedness between two items using the overlap of their sets of inlinks and outlinks in the Wikipedia hyperlink graph.
- *Katz* Based on Katz’s centrality measure, the relatedness between two items is computed as the accumulated probability of the top shortest paths between their corresponding entities in the semantic network (Hulpus et al. 2015).

We evaluated the previous semantic relatedness metrics indirectly by comparing their performance in the item recommendation task. For such purpose, we chose a content-based recommendation model with no parameters, so that we can fairly measure the effect of each similarity on the item ranking quality. According to this simple model, the relevance score of an item is computed as the accumulated similarity with the items in the user’s profile:

$$s(u, i) = \sum_{j \in I(u)} s_{ij} \quad (38)$$

where s_{ij} is computed any of the methods described above.

The results of our experiment are shown in Table 6. For an easier comparison according to the methodology from Sect. 5.2, we averaged the MRR scores for all the cold start sizes in each source–target domain combination. From the table we conclude that M&W is the best performing metric, beating all the other approaches except when considering the movie domain as source, in which case it is still competitive. Hence, in the following experiments we evaluate our proposed matrix factorization models using M&W as the backing semantic similarity. We note that better results could be achieved in real-world systems by selecting the optimal similarity metric for each domain. However, using the same metric simplifies the experiments and can only harm the performance of our proposed models, as the baselines do not exploit such information. Finally, we note that the low values for MRR are due to the simple recommendation algorithm chosen for this experiment.

Table 6 MRR of the evaluated semantic relatedness metrics

Source	Target	TF-IDF	ESA	M&W	Katz
Books	Movies	0.058	0.030	0.123	0.092
	Music	0.028	0.015	0.042	0.022
Movies	Books	0.054	0.011	0.031	0.013
	Music	0.030	0.011	0.028	0.009
Music	Books	0.010	0.006	0.052	0.020
	Movies	0.013	0.018	0.088	0.006

Best values for each row are shown in bold

5.6 Item ranking accuracy

In our second experiment we analyze the accuracy of the item rankings generated by the evaluated recommendation approaches. We aim to understand if cross-domain variants are in general more effective than single-domain ones, and whether the proposed matrix factorization models are able to outperform the other methods in cold start settings.

Table 7 shows the ranking accuracy for book recommendations in terms of MRR. We report the average results for cold start user profiles from sizes 6–10, as we observed that in those cases the trends are stable and, in general, single-domain baselines start to be effective. We remark that, according the evaluation methodology described in Sect. 5.2, the number of test users remains constant regardless of the profile size, which we control by iteratively downsampling the training portion of their profile (see Fig. 1).

We notice from the table that, in general, approaches exploiting cross-domain movies or music preferences provide better recommendations than their single-domain counterparts. In case auxiliary movie preferences are available, we observe that the proposed NeighborMF and CentroidMF models achieve the best performance when only 1–3 book likes are observed (statistically significant with respect to CD-INN, Wilcoxon test, $p < 0.05$). Moreover, in that case, our cross-domain matrix factorization models perform much better than the single-domain baselines. However, once 4 likes are available, CD-INN and single-domain HeteRec are more effective approaches. When the auxiliary preferences consist of music likes, we see that CD-INN is the overall best method, although it is only useful for profiles of size 1. For larger profiles, it is better to use single-domain baselines than any cross-domain method that uses music preferences. In summary, we conclude that music preferences are not useful for book recommendations, whereas movie likes could be used to improve the performance, specially with NeighborMF and CentroidMF for 1–3 book likes. We observe the bad performance of SPRank in cold start situations compared to the other baselines.

In Table 8 we show the results for movie recommendations.

We observe that most of the cross-domain approaches are able to provide recommendations better than the most popular items for completely new movie users, and that CD-HeteRec is clearly the best performing approach. Strangely, its performance starts degrading as soon as some likes become available when it is surpassed by other methods. We have yet to find a clear explanation for this anomaly, which we plan to investigate in the future. If the auxiliary cross-domain data consists of book prefer-

Table 7 Accuracy (MRR) for cold start users in the target books domain. The three groups of rows correspond to single-domain, cross-domain with movies as source, and cross-domain with music as source, respectively. Best values for each single- and cross-domain configuration are shown in bold, and overall best values are underlined

Method	Number of book likes						
	0	1	2	3	4	5	6–10
POP	0.242	0.244	0.246	0.248	0.251	0.252	0.260
UNN		0.222	0.265	0.286	0.289	0.290	0.322
INN		0.145	0.177	0.216	0.241	0.262	0.316
iMF		0.171	0.194	0.235	0.255	0.271	0.301
BPR		0.110	0.116	0.136	0.154	0.157	0.193
FISM		0.228	0.230	0.234	0.234	0.238	0.245
HeteRec		0.218	0.244	0.279	0.297	0.316	0.351
SPRank		0.048	0.055	0.070	0.065	0.062	0.059
Movies							
CD-UNN	0.186	0.148	0.170	0.175	0.189	0.190	0.212
CD-INN	0.262	0.265	0.275	0.291	0.301	0.307	0.339
CD-iMF	0.261	0.262	0.268	0.272	0.275	0.274	0.287
CD-BPR	0.217	0.200	0.218	0.237	0.235	0.238	0.251
CD-FISM	0.235	0.228	0.225	0.231	0.236	0.235	0.245
CD-HeteRec	0.264	0.248	0.261	0.268	0.278	0.277	0.298
SimMF	0.253	0.268	0.274	0.284	0.289	0.290	0.296
NeighborMF	0.253	0.272	0.282	0.294	0.293	0.293	0.301
CentroidMF	0.252	0.271	0.283	0.289	0.293	0.295	0.301
Music							
CD-UNN	0.136	0.103	0.115	0.120	0.138	0.140	0.157
CD-INN	0.259	0.260	0.266	0.278	0.296	0.302	0.329
CD-iMF	0.259	0.261	0.262	0.264	0.266	0.270	0.282
CD-BPR	0.218	0.199	0.199	0.216	0.228	0.228	0.250
CD-FISM	0.230	0.228	0.227	0.229	0.236	0.233	0.245
CD-HeteRec	0.266	0.249	0.251	0.259	0.270	0.267	0.281
SimMF	0.255	0.259	0.258	0.264	0.268	0.273	0.281
NeighborMF	0.253	0.258	0.258	0.263	0.267	0.273	0.280
CentroidMF	0.255	0.259	0.260	0.264	0.267	0.273	0.281

ences, we notice that the proposed matrix factorization models outperform the best single-domain baselines. However, in this situation CD-INN is even a better method, clearly providing more accurate recommendations than any other approach from profile sizes 1–10. This is due to the high degree of overlap between the users of books and movies domains (79.7%, see Table 4), which allows CD-INN to compute very accurate item similarities based on the patterns of likes. Instead, when the source domain contains music preferences, we see that NeighborMF, CentroidMF, and SimMF, in that order, are consistently the best performing approaches for sizes 1–10 (statistically sig-

Table 8 Accuracy (MRR) for cold start users in the target movies domain. The three groups of rows correspond to single-domain, cross-domain with books as source, and cross-domain with music as source, respectively. Best values for each single- and cross-domain configuration are shown in bold, and overall best values are underlined

Method	Number of movie likes						
	0	1	2	3	4	5	6–10
POP	0.285	0.287	0.289	0.292	0.294	0.297	0.305
UNN		0.332	0.320	0.318	0.330	0.348	0.405
INN		0.233	0.300	0.336	0.359	0.377	0.413
iMF		0.256	0.291	0.314	0.334	0.348	0.388
BPR		0.225	0.256	0.276	0.299	0.315	0.350
FISM		0.257	0.265	0.263	0.266	0.267	0.270
HeteRec		0.315	0.346	0.357	0.366	0.374	0.395
SPRank		0.107	0.131	0.139	0.142	0.140	0.150
Books							
CD-UNN	0.219	0.169	0.185	0.219	0.256	0.292	0.385
CD-INN	0.344	0.347	0.371	0.386	0.398	0.410	0.435
CD-iMF	0.267	0.298	0.325	0.347	0.365	0.377	0.413
CD-BPR	0.018	0.189	0.237	0.254	0.278	0.298	0.326
CD-FISM	0.338	0.267	0.263	0.283	0.273	0.287	0.282
CD-HeteRec	0.479	0.320	0.349	0.359	0.367	0.375	0.396
SimMF	0.328	0.334	0.348	0.361	0.371	0.382	0.409
NeighborMF	0.330	0.335	0.348	0.361	0.371	0.383	0.409
CentroidMF	0.329	0.332	0.346	0.359	0.371	0.378	0.408
Music							
CD-UNN	0.387	0.282	0.305	0.320	0.334	0.348	0.383
CD-INN	0.342	0.347	0.353	0.359	0.365	0.371	0.390
CD-iMF	0.301	0.326	0.344	0.362	0.374	0.385	0.418
CD-BPR	0.352	0.305	0.316	0.332	0.332	0.343	0.361
CD-FISM	0.105	0.089	0.093	0.089	0.091	0.091	0.093
CD-HeteRec	0.367	0.336	0.344	0.350	0.355	0.360	0.374
SimMF	0.339	0.351	0.361	0.374	0.384	0.396	0.419
NeighborMF	0.353	0.364	0.374	0.385	0.394	0.404	0.427
CentroidMF	0.345	0.355	0.367	0.377	0.385	0.395	0.418

nificant with respect to CD-INN for sizes 1–2 and CD-iMF for sizes 3–10, Wilcoxon test, $p < 0.05$). By regularizing item factors independently, NeighborMF is able to transfer source domain knowledge more effectively, which we also note is due to the greater contribution of cross-domain information (larger values of λ_C in Table 5). In summary, both book and music preferences are helpful for cold start movie recommendations, while our models are more effective when exploiting auxiliary music likes. On a side note, we observe the better performance of UNN over POP on the single domain setting when only 1 like is available. Looking at the results, we found that this

Table 9 Accuracy (MRR) for cold start users in the target music domain. The three groups of rows correspond to single-domain, cross-domain with books as source, and cross-domain with movies as source, respectively. Best values for each single- and cross-domain configuration are shown in bold, and overall best values are underlined

Method	Number of music likes						
	0	1	2	3	4	5	6–10
POP	0.335	0.337	0.340	0.342	0.345	0.347	0.354
UNN		0.422	0.389	0.389	0.419	0.448	0.517
INN		0.320	0.391	0.426	0.455	0.474	0.517
iMF		0.347	0.396	0.427	0.451	0.471	0.517
BPR		0.330	0.377	0.409	0.432	0.450	0.488
FISM		0.096	0.100	0.100	0.100	0.101	0.100
HeteRec		0.358	0.395	0.421	0.442	0.463	0.510
SPRank		N/A	N/A	N/A	N/A	N/A	N/A
Books							
CD-UNN	0.290	0.244	0.266	0.300	0.344	0.387	0.487
CD-INN	0.310	0.368	0.416	0.442	0.465	0.482	0.522
CD-iMF	0.200	0.330	0.391	0.423	0.451	0.471	0.518
CD-BPR	0.004	0.267	0.323	0.362	0.380	0.404	0.433
CD-FISM	0.153	0.124	0.105	0.126	0.116	0.118	0.113
CD-HeteRec	0.514	0.367	0.407	0.432	0.453	0.474	0.516
SimMF	0.310	0.368	0.401	0.424	0.446	0.461	0.496
NeighborMF	0.328	0.372	0.402	0.425	0.445	0.461	0.496
CentroidMF	0.325	0.370	0.402	0.425	0.444	0.461	0.496
Movies							
CD-UNN	0.435	0.274	0.306	0.336	0.369	0.400	0.484
CD-INN	0.412	0.431	0.451	0.467	0.478	0.490	0.522
CD-iMF	0.293	0.356	0.398	0.428	0.454	0.474	0.516
CD-BPR	0.431	0.313	0.351	0.391	0.402	0.413	0.448
CD-FISM	0.093	0.061	0.069	0.067	0.070	0.071	0.064
CD-HeteRec	0.515	0.406	0.426	0.442	0.451	0.464	0.495
SimMF	0.361	0.393	0.420	0.438	0.455	0.467	0.500
NeighborMF	0.353	0.385	0.409	0.429	0.445	0.458	0.494
CentroidMF	0.354	0.386	0.413	0.431	0.447	0.460	0.495

is caused by the Jaccard-based similarity, which favors neighbors with small profiles that have rated similar items with high probability. A discussion of this phenomenon is outside of the scope of this paper, and we refer the reader to Bellogín et al. (2018) for a detailed explanation. HeteRec, on the other hand, exploits additional information from item metadata to compute more accurate recommendations than POP, while SPRank confirms its bad behavior in cold start scenarios.

Finally, the results for music recommendations are shown in Table 9. As previously, CD-HeteRec is a very good performing approach to provide recommendations

for completely new users in both cross-domain configurations, before significantly dropping for larger profiles. Once 2 music likes are available, CD-INN is clearly the most competitive approach, independently of the used source domain. Again, we argue that this is due to the high number of music users who also have book and movie preferences, which allows CD-INN to compute very accurate rating-based similarities for items (see last column of Table 4). However, when the source domain consists of book preferences, we see that the proposed NeighborMF and CentroidMF models are slightly better than other cross-domain approaches if only 1 music like is provided. Anyway, even better performance can be achieved in this case simply using the single-domain UNN baseline, which does not need any extra information. Hence, single-domain baselines are compelling approaches for cold start music recommendations, and even though the proposed models are able to improve the quality of the item rankings by exploiting cross-domain item metadata, CD-INN, which is purely based on patterns of likes, is the best performing approach.

5.7 Recommendation diversity

In this subsection we analyze the diversity of the recommendation lists generated by the methods, as an alternative dimension of ranking quality.

Table 10 shows the diversity of book recommendations in terms of the Binomial Diversity metric at cutoff 10 (BinomDiv@10). We observe that, in general, cross-domain approaches provide more diverse recommendations than their single-domain counterparts. However, we note several differences with respect to the accuracy results reported in Table 7. First, CD-UNN is consistently the superior algorithm in terms of diversity, whereas its accuracy results were the poorest among single- and cross-domain approaches. Second, when the source domain consists of movie likes, our proposed models achieve slightly worse diversity than other cross-domain approaches, specially for book profile sizes between 1–3 likes. This is in contrast with the results obtained in Table 7, where our methods performed best precisely in that range. We conclude that there is a clear trade-off between recommendation accuracy and diversity, and that the metric of interest depends on the particular application domain. We argue, however, that in cold start situations providing relevant suggestions may be more useful than recommending diverse, but not relevant items, if the ultimate goal of a system is to keep new users engaged.

The diversity results for movie recommendations are summarized in Table 11. We see that CD-FISM, CD-BPR, and CD-UNN provide the most diverse yet not relevant recommendations. Comparing the sources of auxiliary user preferences, we note that the diversity of the cross-domain baselines is roughly the same as their single-domain versions (comparing e.g. HeteRec and CD-HeteRec) when considering book likes. In contrast, if the source domain contains music likes, their diversity is significantly hurt. By comparing these results with Table 8 we observe once again the accuracy-diversity trade-off. Most methods' MRR greatly benefits from additional music likes at the expense of worse diversity. The exception is CD-FISM, which follows the opposite trend: source music likes lead to significantly worse accuracy but improved diversity. We leave for future work an analysis of CD-FISM to understand which of its charac-

Table 10 Diversity (BinomDiv@10) for cold start users in the books domain. The three groups of rows correspond to single-domain, cross-domain with movies as source, and cross-domain with music as source, respectively. Best values for each single- and cross-domain configuration are shown in bold, and overall best values are underlined

Method	Number of book <i>likes</i>						
	0	1	2	3	4	5	6–10
POP	0.739	0.674	0.690	0.702	0.703	0.710	0.736
UNN		0.733	0.706	0.716	0.709	0.729	0.715
INN		0.655	0.674	0.654	0.665	0.672	0.669
iMF		0.583	0.606	0.630	0.645	0.657	0.664
BPR		0.696	0.700	0.715	0.690	0.698	0.696
FISM		0.513	0.692	0.708	0.686	0.706	0.719
HeteRec		0.609	0.623	0.653	0.672	0.680	0.693
SPRank		0.121	0.129	0.145	0.157	0.157	0.150
Movies							
CD-UNN	0.792	0.833	0.816	0.791	0.778	0.784	0.746
CD-INN	0.740	0.676	0.683	0.684	0.680	0.692	0.695
CD-iMF	0.724	0.660	0.674	0.689	0.686	0.686	0.702
CD-BPR	0.514	0.458	0.484	0.425	0.454	0.465	0.471
CD-FISM	0.453	0.464	0.479	0.488	0.495	0.499	0.533
CD-HeteRec	0.747	0.673	0.672	0.680	0.690	0.704	0.709
SimMF	0.702	0.649	0.671	0.676	0.682	0.690	0.706
NeighborMF	0.690	0.652	0.660	0.671	0.680	0.682	0.702
CentroidMF	0.699	0.647	0.659	0.668	0.684	0.686	0.702
Music							
CD-UNN	0.744	0.811	0.797	0.771	0.746	0.734	0.731
CD-INN	0.746	0.676	0.683	0.684	0.674	0.689	0.691
CD-iMF	0.720	0.657	0.664	0.674	0.690	0.692	0.696
CD-BPR	0.303	0.333	0.374	0.384	0.358	0.374	0.380
CD-FISM	0.345	0.362	0.382	0.405	0.406	0.424	0.466
CD-HeteRec	0.744	0.668	0.655	0.665	0.676	0.687	0.693
SimMF	0.724	0.656	0.675	0.684	0.692	0.692	0.708
NeighborMF	0.721	0.657	0.674	0.684	0.690	0.693	0.709
CentroidMF	0.721	0.655	0.673	0.681	0.692	0.690	0.705

teristics causes this behavior. Finally, we remark the good performance of the NeighborMF method when source music likes are exploited, as it is able to provide a good trade-off of decent diversity and the most accurate recommendations (see Table 8).

Last, we report the diversity results for music recommendations in Table 12. Once again, CD-FISM, which achieved the poorest accuracy in Table 9, provides the most diverse recommendations for all music profile sizes in the 1–10 range. However, for completely new users, we highlight the very good performance of CD-HeteRec, which not only is able to generate diverse recommendations, but also achieved the

Table 11 Diversity (BinomDiv@10) for cold start users in the movies domain. The three groups of rows correspond to single-domain, cross-domain with books as source, and cross-domain with music as source, respectively. Best values for each single- and cross-domain configuration are shown in bold, and overall best values are underlined

Method	Number of movie likes						
	0	1	2	3	4	5	6–10
POP	0.401	0.304	0.336	0.354	0.368	0.378	0.399
UNN		0.360	0.385	0.404	0.392	0.396	0.394
INN		0.289	0.308	0.315	0.321	0.323	0.332
iMF		0.299	0.320	0.335	0.344	0.347	0.362
BPR		0.590	0.608	0.628	0.644	0.650	0.653
FISM		0.561	0.614	0.594	<u>0.689</u>	0.636	0.644
HeteRec		0.311	0.328	0.334	0.337	0.341	0.348
SPRank		0.218	0.242	0.260	0.262	0.254	0.269
Books							
CD-UNN	0.467	0.509	0.479	0.446	0.425	0.414	0.397
CD-INN	0.327	0.291	0.314	0.323	0.329	0.331	0.339
CD-iMF	0.341	0.294	0.317	0.327	0.333	0.338	0.350
CD-BPR	0.646	0.677	0.645	<u>0.668</u>	0.624	0.643	0.666
CD-FISM	0.548	0.549	0.671	0.574	0.609	0.625	0.664
CD-HeteRec	0.316	0.310	0.328	0.335	0.337	0.341	0.348
SimMF	0.308	0.265	0.297	0.307	0.320	0.325	0.339
NeighborMF	0.315	0.266	0.298	0.306	0.321	0.325	0.338
CentroidMF	0.313	0.273	0.302	0.315	0.326	0.334	0.348
Music							
CD-UNN	0.368	0.404	0.386	0.376	0.373	0.372	0.376
CD-INN	0.309	0.240	0.268	0.283	0.297	0.304	0.321
CD-iMF	0.270	0.231	0.270	0.289	0.302	0.315	0.332
CD-BPR	0.372	0.439	0.411	0.438	0.446	0.445	0.476
CD-FISM	<u>0.653</u>	<u>0.720</u>	<u>0.705</u>	0.499	0.645	<u>0.732</u>	0.688
CD-HeteRec	0.333	0.271	0.298	0.314	0.324	0.333	0.349
SimMF	0.311	0.254	0.288	0.303	0.317	0.324	0.340
NeighborMF	0.311	0.259	0.290	0.308	0.320	0.329	0.344
CentroidMF	0.302	0.246	0.279	0.297	0.310	0.319	0.338

best accuracy results in terms of MRR. The remaining cross-domain approaches are in general worse than single-domain UNN, independently of the exploited source domain. It is also worth noting the contrasting results for CD-INN. While it provides the best performance in terms of accuracy (see Table 9), its diversity is the worst for books and only average for movies.

In summary, we observe a clear trade-off between accurate and diverse recommendations. In general, when approaches perform well in terms of MRR they tend to suffer in terms of diversity, and vice versa.

Table 12 Diversity (BinomDiv@10) for cold start users in the music domain. The three groups of rows correspond to single-domain, cross-domain with books as source, and cross-domain with movies as source, respectively. Best values for each single- and cross-domain configuration are shown in bold, and overall best values are underlined

Method	Number of music <i>likes</i>						
	0	1	2	3	4	5	6–10
POP	0.324	0.228	0.262	0.282	0.295	0.305	0.326
UNN		0.296	0.332	0.348	0.347	0.330	0.306
INN		0.200	0.213	0.219	0.223	0.229	0.236
iMF		0.196	0.217	0.232	0.241	0.249	0.259
BPR		0.539	0.577	0.589	0.590	0.594	0.619
FISM		0.683	0.766	0.709	0.731	0.737	0.676
HeteRec		0.227	0.264	0.280	0.288	0.296	0.304
SPRank		N/A	N/A	N/A	N/A	N/A	N/A
Books							
CD-UNN	0.325	0.429	0.414	0.393	0.366	0.346	0.314
CD-INN	0.269	0.215	0.227	0.232	0.235	0.240	0.244
CD-iMF	0.270	0.214	0.233	0.240	0.249	0.252	0.258
CD-BPR	0.570	0.585	0.578	0.602	0.592	0.603	0.609
CD-FISM	0.607	0.597	0.677	0.648	0.774	0.726	0.674
CD-HeteRec	0.295	0.233	0.271	0.286	0.294	0.302	0.309
SimMF	0.274	0.220	0.240	0.249	0.257	0.264	0.275
NeighborMF	0.254	0.220	0.241	0.251	0.259	0.265	0.275
CentroidMF	0.253	0.218	0.238	0.249	0.257	0.263	0.273
Movies							
CD-UNN	0.296	0.411	0.380	0.358	0.347	0.329	0.312
CD-INN	0.277	0.231	0.255	0.264	0.270	0.272	0.275
CD-iMF	0.248	0.229	0.254	0.264	0.271	0.272	0.277
CD-BPR	0.476	0.515	0.526	0.504	0.526	0.529	0.545
CD-FISM	0.601	0.664	0.541	0.757	0.578	0.771	0.669
CD-HeteRec	0.372	0.271	0.314	0.331	0.342	0.349	0.360
SimMF	0.225	0.207	0.239	0.250	0.259	0.264	0.278
NeighborMF	0.252	0.226	0.251	0.265	0.269	0.274	0.283
CentroidMF	0.264	0.233	0.257	0.270	0.274	0.279	0.286

6 Conclusions and future work

Collaborative filtering approaches have become the most investigated and popular solutions to the cross-domain recommendation problem, as they only mine patterns of user–item preferences (i.e., ratings), and do not require any information about the content of the items to bridge the domains of interest. Some other approaches, however, have shown that content-based relations (e.g., based on social tags) can be exploited to bridge the domains more effectively. In this context, recent initiatives such

as the Linked Open Data project provide large interconnected repositories of structured knowledge than can be exploited to relate multiple types of data. Such heterogeneous networks allow establishing content-based links between different types of items, and thus providing a new mechanism to bridge domains for cross-domain recommendation.

In this paper, we have exploited Linked Open Data to extract metadata about items in three recommendation domains. Using this additional information, we were able to find relations between items in different domains, and ultimately compute inter-domain item similarities. This could be a limit of the presented approaches whenever the underlying LOD knowledge graph does not expose semantic links between items in different domains, e.g., when the source and target domains do not share information, i.e., there is no direct or indirect link between items in different domains or it is not possible to link an item in the catalog to the corresponding entity in the knowledge graph. In fact, in these cases, it is not possible to compute pairwise semantic similarity values between items belonging to different domains.

We then proposed three novel matrix factorization models for cross-domain recommendation that exploit the computed similarities to link knowledge across domains. Experiments in cold-start scenarios showed that depending on the involved source and target domains, cross-domain recommendations exploiting item metadata can be more accurate for users with few preferences in the target domain. However, the improved accuracy comes at the cost of less diversity among the recommendations, and baseline approaches thriving in diversity tend to be less accurate. We argue, nonetheless, that in cold start the priority of a system may be keeping the user engaged by delivering relevant recommendations rather than diverse, non relevant ones.

Regarding the categorization presented in Sect. 2.1, the models proposed in this paper belong to the category of *knowledge linkage* cross-domain recommendation approaches. We applied our approaches to the *linked-domain exploitation* task with the goal of *addressing the user cold-start* problem. In addition to the results reported in this paper, we conjecture that item metadata may prove more useful in cross-domain scenarios with low user overlap. In these cases, approaches purely based on collaborative filtering are likely to struggle to compute accurate item-item similarities. Moreover, in our work we relied on advanced Bayesian Optimization techniques to find the optimal hyper-parameters of the models, and in particular the values of the cross-domain regularization λ_C and the item neighborhood size n parameters. It would be interesting, however, to analyze the performance of the models in terms of these parameters to better understand the importance of auxiliary information. We did not report these results in the paper due to the high number of possible combinations of different parameter values, source–target domain configurations, cold start profile sizes, and cross-validation folds, which may make it very difficult to extract conclusions that consistently hold through all the possible scenarios.

Finally, the work presented in this paper can be extended in future contributions by improving the experimental section, e.g., considering more cross-domain datasets linked to DBpedia or adapting more novel approaches from the state of the art such as those based on tensor factorization (Taneja and Arora 2018) and deep learning (Zhu et al. 2018).

Acknowledgements This work was supported by the Spanish Ministry of Economy, Industry and Competitiveness (TIN2016-80630-P). The authors thank the reviewers for their thoughtful comments and suggestions.

References

- Abel, F., Araújo, S., Gao, Q., Houben, G.J.: Analyzing cross-system user modeling on the social web. In: Proceedings of the 11th International Conference on Web Engineering (ICWE'11), pp. 28–43 (2011)
- Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. *User Model. User Adapt. Interact.* **23**(2–3), 169–209 (2013)
- Bell, R.M., Koren, Y.: Lessons from the Netflix prize challenge. *SIGKDD Explor.* **9**(2), 75–79 (2007a)
- Bell, R.M., Koren, Y.: Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: Proceedings of the 7th IEEE International Conference on Data Mining (ICDM'07), pp. 43–52 (2007b)
- Bellofín, A., Fernández-Tobías, I., Cantador, I., Tomeo, P.: Neighbor selection for cold users in collaborative filtering with positive-only feedback. In: Proceedings of the 18th Conference of the Spanish Association for Artificial Intelligence (CAEPIA'18), pp. 3–12 (2018)
- Berkovsky, S., Kuflik, T., Ricci, F.: Distributed collaborative filtering with domain specialization. In: Proceedings of the 1st ACM Conference on Recommender Systems (RecSys'07), pp. 33–40 (2007a)
- Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of user models for enhanced personalization in recommender systems. *User Model. User Adapt. Interact.* **18**(3), 245–286 (2007b)
- Bizer, C., Heath, T., Berners-Lee, T.: Linked data—the story so far. *Int. J. Semant. Web Inf. Syst.* **5**(3), 1–22 (2009)
- Cantador, I., Fernández-Tobías, I., Bellofín, A.: Relating personality types with user preferences in multiple entertainment domains. In: Proceeding of the 1st International Workshop on Emotions and Personality in Personalized Services (EMPIRE'13) (2013)
- Cantador, I., Fernández-Tobías, I., Berkovsky, S., Cremonesi, P.: Cross-domain recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) *Recommender Systems Handbook*, pp. 919–959. Springer, Berlin (2015)
- Cao, B., Liu, N.N., Yang, Q.: Transfer learning for collective link prediction in multiple heterogenous domains. In: Proceedings of the 27th International Conference on Machine Learning (ICML'10), pp. 159–166 (2010)
- Chung, R., Sundaram, D., Srinivasan, A.: Integrated personal recommender systems. In: Proceedings of the 9th International Conference on Electronic Commerce (ICE'07), pp. 65–74 (2007)
- Cremonesi, P., Quadrana, M.: Cross-domain recommendations without overlapping data: myth or reality? In: Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14), pp. 297–300 (2014)
- Cremonesi, P., Tripodi, A., Turrin, R.: Cross-domain recommender systems. In: Proceedings of the 11th IEEE International Conference on Data Mining Workshops (ICDMW'11), pp. 496–503 (2011)
- Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **41**(6), 391–407 (1990)
- Di Noia, T., Ostuni, V.C., Tomeo, P., Di Sciascio, E.: Sprank: semantic path-based ranking for top-n recommendations using Linked Open Data. *ACM Trans. Intell. Syst. Technol.* **8**(1), 9 (2016)
- Enrich, M., Braunhofer, M., Ricci, F.: Cold-start management with cross-domain collaborative filtering and tags. In: Proceedings of the 14th International Conference on E-Commerce and Web Technologies (EC-Web'13), pp. 101–112 (2013)
- Fernández-Tobías, I., Cantador, I.: Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. In: Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems (CBRecSys'14), pp. 34–41 (2014)
- Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.: A generic semantic-based framework for cross-domain recommendation. In: Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec'11), pp. 25–32 (2011)
- Fernández-Tobías, I., Cantador, I., Plaza, L.: An emotion dimensional model based on social tags: crossing folksonomies and enhancing recommendations. In: Proceedings of the 14th International Conference on E-Commerce and Web Technologies (EC-Web'13), pp. 88–100 (2013)

- Funk, S.: Netflix update: try this at home. <http://sifter.org/~simon/journal/20061211.html> (2006). Accessed 12 Dec 2017
- Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), pp. 1606–1611 (2007)
- Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J.: Cross-domain recommendation via cluster-level latent factor model. In: Proceedings of 23th and 16th European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD'13), pp. 161–176 (2013)
- Givon, S., Lavrenko, V.: Predicting social-tags for cold start book recommendations. In: Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys'09), pp. 333–336 (2009)
- Goga, O., Lei, H., Parthasarathi, S.H.K., Friedland, G., Sommer, R., Teixeira, R.: Exploiting innocuous activity for correlating users across sites. In: Proceedings of the 22nd International World Wide Web Conference (WWW'13), pp. 447–458 (2013)
- Guo, G., Zhang, J., Sun, Z., Yorke-Smith, N.: Librec: A Java library for recommender systems. In: Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP'15) (2015)
- He, M., Zhang, J., Yang, P., Yao, K.: Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM'18), pp. 225–233 (2018)
- Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 230–237 (1999)
- Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Zhu, C.: Personalized recommendation via cross-domain triadic factorization. In: Proceedings of the 22nd International World Wide Web Conference (WWW'13) pp. 595–605 (2013)
- Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08), pp. 263–272 (2008)
- Hulpus, I., Prangnawarat, N., Hayes, C.: Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In: Proceedings of the 14th International Semantic Web Conference (ISWC'15), pp. 442–457 (2015)
- Jain, P., Kumaraguru, P., Joshi, A.: @i seek 'fb.me': identifying users across multiple online social networks. In: Proceedings of the 22nd International World Wide Web Conference (WWW'13), pp. 1259–1268 (2013)
- Kabbur, S., Ning, X., Karypis, G.: FISM: Factored item similarity models for top-n recommender systems. In: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13), pp. 659–667 (2013)
- Kaminskas, M., Fernández-Tobías, I., Ricci, F., Cantador, I.: Ontology-based identification of music for places. In: Proceedings of the 13th International Conference on Information Communication Technologies in Tourism (ENTER'13), pp. 436–447 (2013)
- Kluser, D., Konstan, J.A.: Evaluating recommender behavior for new users. In: Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14), pp. 121–128 (2014)
- Koren, Y.: Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08), pp. 426–434 (2008)
- Koren, Y., Bell, R.M.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 77–118. Springer, Berlin (2015)
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web J.* **6**(2), 167–195 (2015)
- Li, B., Yang, Q., Xue, X.: Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09), pp. 2052–2057 (2009a)
- Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09), pp. 617–624 (2009b)
- Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput.* **7**(1), 76–80 (2003)

- Liu, J., Shi, J., Cai, W., Liu, B., Pan, W., Yang, Q., Ming, Z.: Transfer learning from APP domain to news domain for dual cold-start recommendation. In: Proceedings of the 1st Workshop on Intelligent Recommender Systems by Knowledge Transfer & Learning (RecSysKTL'17), pp. 38–41 (2017)
- Loni, B., Shi, Y., Larson, M., Hanjalic, A.: Cross-domain collaborative filtering with factorization machines. In: Proceedings of the 36th European Conference on IR Research (ECIR'14), pp. 656–661 (2014)
- Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceedings of the 2008 AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, pp. 25–30 (2008)
- Mirbakhsh, N., Ling, C.X.: Improving top-n recommendation for cold-start users via cross-domain information. *ACM Trans. Knowl. Discov. Data* **9**(4), 33:1–33:19 (2015)
- Pan, W.: A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing* **177**(C), 447–453 (2016)
- Pan, W., Xiang, E.W., Liu, N.N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10), pp. 210–235 (2010)
- Pan, W., Liu, N.N., Xiang, E.W., Yang, Q.: Transfer learning to predict missing ratings via heterogeneous user feedbacks. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11), pp. 2318–2323 (2011)
- Paquet, U., Koenigstein, N.: One-class collaborative filtering with random graphs. In: Proceedings of the 22nd International World Wide Web Conference (WWW'03), pp. 999–1008 (2013)
- Pilászky, I., Zibriczky, D., Tikk, D.: Fast ALS-based matrix factorization for explicit and implicit feedback datasets. In: Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10), pp. 71–78 (2010)
- Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09), pp. 452–461 (2009)
- Rowe, M.: SemanticSVD++: Incorporating semantic taste evolution for predicting ratings. In: Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT'14), pp. 213–220 (2014)
- Sahebi, S., Brusilovsky, P.: Cross-domain collaborative recommendation in a cold-start context: the impact of user profile size on the quality of recommendation. In: Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization (UMAP'13), pp. 289–295 (2013)
- Sahebi, S., Brusilovsky, P., Bobrovok, V.: Cross-domain recommendation for large-scale data. In: Proceedings of the 1st Workshop on Intelligent Recommender Systems by Knowledge Transfer & Learning (RecSysKTL'17), pp. 9–15 (2017)
- Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS'07), pp. 1257–1264 (2007)
- Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of dimensionality reduction in recommender system—a case study. In: Proceedings of the ACM WebKDD-2000 Workshop (2000)
- Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. *IEEE Intell. Syst.* **21**(3), 96–101 (2006)
- Shapira, B., Rokach, L., Freilikhman, S.: Facebook single and cross domain data for recommendation systems. *User Model. User Adapt. Interact.* **23**(2–3), 211–247 (2013)
- Shi, Y., Larson, M., Hanjalic, A.: Tags as bridges between domains: improving recommendation with tag-induced cross-domain collaborative filtering. In: Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization (UMAP'11), pp. 305–316 (2011)
- Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS'12), pp. 2960–2968 (2012)
- Stewart, A., Diaz-Aviles, E., Nejdil, W., Marinho, L.B., Nanopoulos, A., Schmidt-Thieme, L.: Cross-tagging for personalized open social networking. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia (Hypertext'09), pp. 271–278 (2009)
- Szomszor, M., Alani, H., Cantador, I., O'Hara, K., Shadbolt, N.: Semantic modelling of user interests based on cross-folksonomy analysis. In: Proceedings of the 7th International Semantic Web Conference (ISWC'08), pp. 632–648 (2008a)
- Szomszor, M., Cantador, I., Alani, H.: Correlating user profiles from multiple folksonomies. In: Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (Hypertext'08), pp. 33–42 (2008b)

- Takács, G., Tikk, D.: Alternating least squares for personalized ranking. In: Proceedings of the 6th ACM Conference on Recommender Systems (RecSys'12), pp. 83–90 (2012)
- Taneja, A., Arora, A.: Cross domain recommendation using multidimensional tensor factorization. *Expert Syst. Appl.* **92**(C), 304–316 (2018)
- Tiroshi, A., Kuflik, T.: Domain ranking for cross domain collaborative filtering. In: Proceedings of the 20th International Conference on User Modeling, Adaption and Personalization (UMAP'12), pp. 328–333 (2012)
- Tiroshi, A., Berkovsky, S., Káafar, M.A., Chen, T., Kuflik, T.: Cross social networks interests predictions based on graph features. In: Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13), pp. 319–322 (2013)
- Vargas, S., Baltrunas, L., Karatzoglou, A., Castells, P.: Coverage, redundancy and size-awareness in genre diversity for recommender systems. In: Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14), pp. 209–216 (2014)
- Winoto, P., Tang, T.: 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 Gener. Comput.* **26**, 209–225 (2008)
- Wongchokprasitti, C., Peltonen, J., Ruotsalo, T., Bandyopadhyay, P., Jacucci, G., Brusilovsky, P.: User model in a box: cross-system user model transfer for resolving cold start problems. In: Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization (UMAP'15), pp. 289–301 (2015)
- Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., Han, J.: Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM'14), pp. 283–292 (2014)
- Zhang, Y., Cao, B., Yeung, D.Y.: Multi-domain collaborative filtering. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI'10), pp. 725–732 (2010)
- Zhang, Q., Wu, D., Lu, J., Liu, F., Zhang, G.: A cross-domain recommender system with consistent information transfer. *Decis. Support Syst.* **104**, 49–63 (2017)
- Zhao, L., Pan, S.J., Yang, Q.: A unified framework of active transfer learning for cross-system recommendation. *Artif. Intell.* **245**, 38–55 (2017)
- Zhu, F., Wang, Y., Chen, C., Liu, G., Orgun, M., Wu, J.: A deep framework for cross-domain and cross-system recommendations. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18), pp. 3711–3717 (2018)
- Zhuang, F., Luo, P., Xiong, H., Xiong, Y., He, Q., Shi, Z.: Cross-domain learning from multiple sources: a consensus regularization perspective. *IEEE Trans. Knowl. Data Eng.* **22**(12), 1664–1678 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ignacio Fernández-Tobías is a data scientist at NTENT in Barcelona, Spain. He received his Ph.D. in Computer Science from Universidad Autónoma de Madrid, Spain, where he worked on matrix factorization models for cross-domain recommendation in user cold-start situations. He has co-authored several papers on cross-domain and context-aware recommendation, and has served as a reviewer for conferences and journals such as ACM IUI and ACM TiiS. His other research interests include model-based collaborative filtering and semantic-based recommendation.

Iván Cantador is a Senior Lecturer of Computer Science at Universidad Autónoma de Madrid (UAM), Spain. He received a Ph.D. degree in Computer Science in 2008 at UAM. After earning his Ph.D., he worked as a research associate at University of Glasgow, UK, and as a postdoctoral research visitor at the Free University of Bolzano (Italy). His current research interests fall in the areas of Recommender Systems, Personalized Information Retrieval, and User Modeling. Dr. Cantador has published over 70 conference and journal papers, and serves regularly as reviewer and scientific committee member in venues of the aforementioned fields, including prestigious conferences such as RecSys, SIGIR, ECIR, WWW, CIKM, Hypertext and UMAP, to name a few.

Paolo Tomeo is currently working as a Data Engineer for Agile Lab in Milan, Italy. He received a Ph.D. in Electrical and Computer Engineering from the Polytechnic University of Bari, Italy. His research interests fall in the areas of Recommender Systems, Machine Learning and Semantic Web. He has published almost 20 scientific papers for international conferences and journals and has served as reviewer for many conferences and committee member for RecSys 2017 and 2018.

Vito Walter Anelli is a Ph.D. student in Electrical and Information Engineering at Polytechnic University of Bari, Italy. His current research interests fall in the areas of Recommender Systems, Knowledge representation, and User Modeling. He has published his works in national and international workshops and conferences as well as in international journals, and served as reviewer in venues of the aforementioned fields, including prestigious journals such as Knowledge-Based Systems, IEEE Transactions on Knowledge and Data Engineering.

Tommaso Di Noia is Full Professor of Information and Data Management with the Polytechnic University of Bari, Italy. Over the years, he has authored several papers in international journals and prominent conferences in his research areas. His current research interests include Recommender systems and machine learning, knowledge graphs and Semantic Web technologies, personalized information access, and preference representation and reasoning.

Affiliations

Ignacio Fernández-Tobías¹ · Iván Cantador¹ · Paolo Tomeo² · Vito Walter Anelli² · Tommaso Di Noia²

✉ Ignacio Fernández-Tobías
ignacio.fernandezt@uam.es

Iván Cantador
ivan.cantador@uam.es

Paolo Tomeo
paolo.tomeo@poliba.it

Vito Walter Anelli
vitowalter.aneli@poliba.it

Tommaso Di Noia
tommaso.dinoia@poliba.it

¹ Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Madrid, Spain

² Dipartimento di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari, Bari, Italy