Leveraging Multiple Networks for Author Personalization

Rohit Parimi and **Doina Caragea**
CIS Department, Kansas State University
Manhattan, Kansas, USA

Abstract

Recommender systems provide personalized item suggestions by identifying patterns in past user-item preferences. Most existing approaches for recommender systems work on a single domain, i.e., use user preferences from one domain and recommend items from the same domain. Recently, some recommendation models have been proposed to use user preferences from multiple related item source domains to improve recommendation accuracy for a target item domain, an area of research known as cross-domain recommender systems. One typical assumption in these systems is that users, items, and user preferences for items are similar across domains. In this paper, we introduce a new cross-domain recommendation problem which does not meet this typical assumption. For example, for some scientometric datasets, when the objective is to recommend co-authors, conferences, and references, respectively, to authors, although the users are similar across domains, the items and user-item preferences are different. To address this problem, we propose two approaches to aggregate knowledge from multiple domains. Our approaches allow us to control the knowledge transferred between domains. Experimental results on a *DBLP* subset show that the proposed cross-domain approaches are helpful in improving recommendation accuracy as compared to single domain approaches.

1 Introduction

The digital revolution over the last decade has resulted in an explosive growth of web information. The rapid increase in online content requires effective and efficient organization of information. This is particularly true for archives of scientific articles with new articles from several proceedings placed online everyday. While this growth has allowed researchers to quickly access more scientific information, it has also made it more difficult for them to find interesting 'articles to read', 'relevant conferences to publish', and 'people to collaborate with'. Recommender systems can be used to address this information overload problem.

Majority of the recommender systems use user preferences for items (*explicit* or *implicit*) from one domain and recommend unknown items from the same domain. For example, Netflix suggests movies, by analyzing existing user ratings for movies and Last.FM recommends artists by analyzing existing implicit user preferences for artists. However, in some real-world datasets, users interact with items from multiple domains and user preferences from these domains, referred to as *source domains*, can be aggregated to improve recommendation accuracy in a *target domain* (known as cross-domain recommender systems). For example, in scientometric datasets, users have interactions with other users, conferences, and references.

More specifically, a scientometric dataset can be modeled as a heterogeneous *implicit* feedback dataset with the following interactions: authors collaborating with other authors form a co-author network[1], authors publishing in conferences form a conference network, and authors referencing papers form a reference network. With the existence of many archives for scientific articles and numerous conferences, authors can benefit from suggestions about possible collaborations, interesting references, and conferences to publish. Recommender systems are ideal to address this problem.The most straightforward way to generate recommendations about co-authors, conferences, and references for an author is to ignore the dependency between user interactions in different networks and treat each network as independent. However, this approach leads to loss of information as networks are related, i.e., information about co-authors of an author can be useful in recommending conferences and references for that author. Similarly, new collaborations can be formed between two authors based on their mutual interest in conferences or references. In this work, we study approaches to combine user interactions is multiple source domains in scientometric datasets to improve target recommendation accuracy.

Most existing research on cross-domain recommender systems is based on latent factor model approaches (Pan et al. 2010), (Li, Yang, and Xue 2009a; 2009b), (Singh and Gordon 2008), although there are some neighborhood approaches to address this problem (Winoto and Tang 2008). However, the underlying assumption in most of these works is that users, items, and user preferences for items are similar in related domains, whereas, for some datasets, although

[1]We use *network* and *domain* interchangeably in this article.

the users are similar across domains, items may not be similar and the related domains do not necessarily share a common preference pattern (Gao et al. 2008). For example, related works on cross-domain recommender systems have simulated a cross-domain framework by using different movie rating datasets as different domains (Pan et al. 2010). Other works have transferred user rating knowledge between movies and books (as they have similarity in genre and there are many movies based on books) (Li, Yang, and Xue 2009a; 2009b). In the case of scientometric datasets, when trying to recommend co-authors, conferences, and references, although the users are similar across domains, items and author preferences for items are different across domains, i.e., authors do not prefer co-authors and conferences in the same way. Furthermore, some methods assumed explicit user preferences in source and target domains (Winoto and Tang 2008), (Li, Yang, and Xue 2009a; 2009b), while others relaxed this assumption for the source domains and considered implicit preferences as Boolean values for them (Pan et al. 2010). In practice, it is easier to find both source and target domains with implicit feedback as this type of data is more common in real-world.

In this work, we propose two approaches that work with implicit feedback data, to aggregate knowledge from source and target domains with similar users but different items, in the context of a neighborhood-based approach to improve recommendation accuracy in the target domain. The proposed approaches handle the aforementioned problem which does not fit with the assumption that items and user-item preferences are similar across domains. In the first approach, we construct a neighborhood for the target domain by aggregating the source and target neighborhoods and use it with a neighborhood approach to recommend items. In the second approach, we aggregate recommendations computed from the source and target domains using a neighborhood approach to generate recommendations. We choose neighborhood approaches because of their intuitiveness and ability to explain recommendations to users as explanations provide transparency and increase user trust in the system (Herlocker, Konstan, and Riedl 2000). Specifically, we used the Adsorption algorithm proposed by Baluja et al. (2006). To summarize, the contributions of this paper are as follows:

- We introduce a new cross-domain recommendation problem that does not meet the assumptions in the literature.
- We propose two ways to use knowledge from multiple domains in the context of a neighborhood-based approach.
- We experiment on a subset of the *DBLP* dataset (to the best of our knowledge, the largest dataset used to date for cross-domain approaches) and show that the proposed approaches are successful in improving recommendation accuracy in the target domain.

The rest of the paper is structured as follows. We review related work on CF approaches and introduce existing research on cross-domain approaches in Section 2. Section 3 explains the Adsorption algorithm and introduces the proposed approaches to capture knowledge from multiple implicit feedback domains. In Sections 4 and 5, we describe the experiments, and results, respectively. Finally, we conclude this work with possible future directions in Section 6.

2 Related Work

In this section, we review popular works on CF approaches and existing cross-domain recommender systems.

Collaborative Filtering: CF has been a popular approach to recommender systems; it is widely used in the past decade to address various problems, regardless of the application domain. These approaches are commonly implemented as neighborhood-based approaches also known as kNN (Sarwar et al. 2001), (Baluja et al. 2006). For more details, the author is referred to the surveys by Desrosiers and Karypis (2011) and Adomavicius and Tuzhilin (2011).

Adsorption: The Adsorption algorithm was proposed by Baluja et al. (2006) for recommending YouTube videos to users by employing random-walks on a user-video graph. This algorithm is a very general semi-supervised framework for classification and works by propagating preference information through the graph structure. The algorithm was successfully used for other tasks such as classification and sentiment analysis (Talukdar and Crammer 2009).

Neighborhood-based Cross-Domain Approaches: The work by Winoto and Tang (2008) is one of the first attempts to use user preferences from multiple domains for recommender systems. The authors conducted several experiments with different combinations of source and target domains with an assumption that all domains contribute equally to the recommendation problem in the target domain and concluded that cross-domain recommendations tend to be less precise than single-domain recommendations. We believe that the source and target domains do not contribute equally to the recommendation problem given the differences in items and user preferences for items across domains and provide the flexibility to control the knowledge transfer.

Latent Factor Model-based Cross-Domain Approaches: Singh and Gordan (2008) proposed Collective Matrix Factorization (CMF) to take advantage of user data in multiple domains. The proposed approach collectively learns from multiple data domains by jointly factorizing the rating matrices and sharing the user latent factors across domains. The underlying assumption is that user preferences are similar across domains.

Li, Yang, and Xue (2009a) present a transfer learning approach that mitigates sparsity in the target rating matrix by using information from a dense source rating matrix. The proposed approach factorizes the source matrix to identify cluster-level rating pattern referred to as the codebook. The codebook is then expanded in the target domain assuming that both source and target domains share the same cluster-level rating patterns and have similar items.

Pan et al. (2010) proposed another transfer learning approach based on matrix factorization for reducing the rating sparsity in target matrix. In this approach, the authors assume the existence of two auxiliary matrices, one with similar users and other with similar items to the target domain but relax the assumption that source and target matrices have homogeneous preferences. The model factorizes the auxiliary user and item matrices to find latent factors for users and items, respectively, and integrate these latent factors into the target rating matrix through a regularized factorization.

We distinguish our work from the above approaches on three points: (i) we do not assume similar items or similar rating patterns for items across domains; (ii) our approach lets us transfer knowledge from multiple source domains with similar users but different items; (iii) we consider implicit feedback for both source and target domains.

3 Problem Formulation and Approaches

In our problem setting, we have a target domain **T** where we address the recommendation problem. In addition, we also have one or more source domains $\mathbf{S}^i$, $i \in [1, n]$ which share the same users (but not the items) with the target domain. Our objective is to make use of user preferences for items in the source domains $\mathbf{S}^i$, in addition to the target domain **T**, to improve target recommendation accuracy. We note that although the source and the target domains are related, items and user preferences for items are different across domains.

To achieve our objective, we propose two approaches[2] that use user preferences in conjunction with a neighborhood-based approach, specifically, Adsorption. In the rest of the section, we will first explain the Adsorption algorithm and later describe the proposed approaches.

3.1 Adsorption Algorithm

The Adsorption algorithm proposed by Baluja et al. propagates user preferences for items on a graph. The intuition behind the algorithm is, for a user, items commonly preferred by similar users are likely to match the user's interests.

Basic Terminology: Let $G = (V, E, w)$ be an undirected graph, where V is the set of users, E is the set of edges between users, and w is the weight on edges. Let L be the set of possible labels and let m be the size of the set L, i.e. $m = |L|$. In a classification setting, labels correspond to classes; in a recommendation setting, labels correspond to items preferred by users in the dataset.

Each user v in the graph is associated with two row-vectors, $y_v, \widehat{y}_v \in \mathbb{R}^m_+$. Vector y_v denotes initial label distribution for user v, i.e., y_{vx} represents the probability that user v prefers label x. Vector $\widehat{y}_v$ indicates predictions made by the algorithm for user v, and encodes a distribution over the m labels. Matrices $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$ indicate the initial label distribution and the algorithm predictions for all users, respectively. The higher the value of y_{vx}, the stronger the belief that user v has a high preference for label x. Similarly, the higher the value of $\widehat{y}_{vz}$, the stronger the a posteriori belief that z corresponds to a good label for user v, assuming that z is a label that was not preferred by the user v a priori, i.e., $y_{vz} = 0$. Using the above definitions, we explain the algorithm using the 'Random-walk View' (see Baluja et al. (2006), Talukdar and Crammer (2009) for more details).

[2] The proposed approaches work in a similar way even when there is only a partial overlap between users in different domains.

Adsorption via Random-walk: The Adsorption algorithm can be described as a random-walk on G. At each node, the algorithm is presented with three options: stop and return, in other words *inject* the initial label distribution y_v of the node, *terminate* or abandon the walk and return an all-zero vector, $\mathbf{0}_m$, or *continue* the walk to neighbor node u chosen according to the probability $Pr[u|v]$, given by Equation (1), and emit predicted labels $\widehat{y}_u$, given by Equation (2). The injection, termination and continuation steps have probabilities p_{inj}, p_{term}, and p_{cont}, respectively, and the sum of these probabilities should be 1. For a particular problem, these probabilities can be selected using cross-validation.

The probability distribution over the neighbors u of a user v is estimated using the following equation:

$$Pr[u|v] = \begin{cases} \dfrac{w_{uv}}{\sum\limits_{u:(u,v)\in E} w_{uv}}, & \text{if } (u,v) \in E \\ 0, & otherwise \end{cases} \tag{1}$$

Furthermore, new labels $\widehat{y}_v$ for a user v can be computed using the following equation:

$$\widehat{y}_v = p_{inj} \times y_v + p_{cont} \times \sum_{u:(u,v)\in E} Pr[u|v]\, \widehat{y}_u + p_{term} \times \mathbf{0}_m \tag{2}$$

The random-walk process is initiated at every node v in the graph G and is repeated until the algorithm converges (i.e., the values in $\widehat{y}_v$ don't change anymore). The final values in $\widehat{y}_v$ are used to make recommendations to the user v. Specifically, items z that have high probability in $\widehat{y}_v$ and have not yet been preferred by user v are recommended.

Neighborhood Construction. Theoretically, in the above algorithm, item preferences for a user v can be propagated to all neighbors of v. However, using all neighbors as opposed to k nearest neighbors is computationally expensive and does not always yield huge improvements in the recommendation accuracy (Bell and Koren 2007), (Desrosiers and Karypis 2011). Thus, in this work, we restrict the random-walk from a user v to only its nearest neighbors u. To compute the nearest neighborhood (kNN), we first compute the weight between all pairs of users and then use the weight to select k nearest neighbors. Based on prior knowledge on implicit feedback datasets, we define weight between two users u and v as the number of common items between the two, normalized by the sum of total items each user preferred, i.e., $w_{uv} = \frac{\#common\ items(u,v)}{\#items(u) + \#items(v)}$. The computed kNN is used with Adsorption to generate user recommendations[3].

3.2 Integrating Multiple Domains

We describe the approaches we propose assuming two sources and one target domain given that the *DBLP* dataset used in this work has three domains. However, the approaches can be generalized to any number of source domains and one target domain.

[3] Some neighbors of a user might have the same weight. To resolve ties, we randomly select n users that have the same weight.

Weighted Aggregation of Neighborhoods (WAN): In each domain, we compute the weight between every pair of users to capture user-user similarity in that domain. The neighborhoods from each domain are then integrated into a single neighborhood for the target domain using a linear combination as shown in Equation (3):

$$WN_T = \alpha N_T + \beta N_{S^1} + \gamma N_{S^2} \tag{3}$$

In the above equation, $N_{S^i}, i = 1, 2$ and N_T are the user neighborhoods for source domains and the target domain, respectively, and WN_T is the weighted neighborhood for the target domain. The parameters α, β, γ control the amount of knowledge from the target and the two source domains, respectively. We use WN_T to compute kNN_T and use the nearest neighbors with the Adsorption algorithm to generate recommendations ($\widehat{\mathbf{Y}}_{\mathbf{T}}$) for the target domain.

Weighted Aggregation of Recommendations (WAR): One problem with the WAN approach is that, for some users, the kNN from WN_T and the kNN from N_T might be the same with different user-user weights. Consequently, for those users, no new information about the neighborhood is captured by using knowledge from multiple domains. To achieve diversity in recommendations for such users, in this approach, we use user-item preferences from each domain to construct neighborhoods, and further use the neighborhoods to recommend target items. Intuitively, the kNN from a source domain also captures user-user similarity and we can replace the kNN from target domain with the kNN from a source domain in Adsorption to recommend target domain items. For example, to recommend collaborations, in addition to propagating co-author preferences using kNN from co-author network, we also propagate co-author preferences using kNN from conference and reference networks. The final recommendations for the target domain are generated by computing a weighted average of recommendations from source and target domains as shown in Equation (4):

$$\begin{aligned}\widehat{\mathbf{Y}}_{\mathbf{T}} = \alpha AS(kNN_T, \mathbf{Y_T}) + \beta AS(kNN_{S^1}, \mathbf{Y_T}) \\ + \gamma AS(kNN_{S^2}, \mathbf{Y_T})\end{aligned} \tag{4}$$

In the above equation, $\widehat{\mathbf{Y}}_{\mathbf{T}}$ corresponds to the final recommendations in the target domain, $AS(kNN_T, \mathbf{Y_T})$ is the set of recommendations from Adsorption using kNN_T and user preferences ($\mathbf{Y_T}$) from target , $AS(kNN_{S^i}, \mathbf{Y_T}), i = 1, 2$ is the set of recommendations from Adsorption using kNN_{S^i} from source domain i and propagating user preferences ($\mathbf{Y_T}$) from target domain. The parameters α, β, γ control the amount of knowledge from the target and the two source domains, respectively. We have to note that the WAR approach is computationally expensive than the WAN approach as we have to run the Adsorption algorithm n (number of source domains) times more in the case of the WAR approach as compared to the WAN approach.

4 Experimental Design

Dataset Description and Preprocessing: We use a dataset downloaded from ArnetMiner[4] (Tang et al. 2008) to construct three domains: a co-author domain in which each tuple has *authorID, coauthorID, #papersCoauthored* information, a conference domain in which each tuple has *authorID, conferenceID, #papersPublished* information, and a reference domain in which each tuple has *authorID, referenceID, #papersReferenced* information.

The original dataset has approximately 2×10^7 publications and 4×10^7 citation relations. From this set, papers published between the years 1990 and 2006 are used to create a training paper set (papers from which information about authors, conferences, and references is extracted to create training data for the three domains) and papers published after the year 2007 are used to create a test paper set (papers from which information about authors, conferences, and references is extracted to create test data for the three domains). Our objective is to recommend collaborators, conferences, and references to authors. We use the following rules to decide if an author will be included in the domains:

1. The author has at least one paper in the training paper set and at least one paper in the test paper set.
2. The author co-authored with at least five different authors in the training paper set and co-authored with at least one author (different to the co-authors from the training paper set) in the test paper set.
3. The author has at least five unique references from all his publications in the training paper set and has at least one reference (different to the references from the training paper set) from all his publications in the test paper set.
4. The author has at least one conference from all his publications in the training paper set and has at least one conference (different to the conference from the training paper set) from all his publications in the test paper set.

After filtering the authors as described above, we are left with $29,189$ authors. For the selected authors, the publications in the training paper set and the test paper set are used to construct the training and the test data for the co-author, conference, and reference domains. We have to note that the co-author, conference, and reference domains we construct have the same users ($29,189$ in each). The training set for the co-author, conference, and reference domains have $140,091$ items (co-authors), $2,393$ items (conferences), and $201,570$ items (references), respectively.

Generally, a user in the test set needs to be present also in the training set (i.e., have some history available) in order to recommend items for this user. The rules described above ensures that this constraint is satisfied. Moreover, given that we want to recommend new unseen items (co-authors or conferences or references) to a user, in each domain, we also remove from the test set of a user items that have already been seen in his/her training set. This is a standard way of creating test data for evaluating recommender systems (Cremonesi, Koren, and Turrin 2010). The filtered test

[4] http://arnetminer.org/citation

set for each domain is then represented as *authorID, ListOfItemsPreferred*, where *ListOfItemsPreferred* is a set of items sorted in descending order based on the preference counts.

Experiments: We aim to understand whether additional knowledge gained from user preferences in auxiliary source domains can be effectively used to improve recommendation accuracy in target domain. Towards this goal, we run three experiments. In each experiment, one domain is used as the target domain and the other two domains as the source domains. We compute Adsorption performance using weighted aggregation of neighborhoods (WAN) and weighted aggregation of recommendations (WAR) approaches and compare the performance from these approaches with the Adsorption performance using just the target domain (our baseline).

Note that in our experiments, for a train user, we randomly pick 50% of preferences from his/her training data and use only these preferences to generate recommendations. This is repeated five times to account for variation in results from the approaches and the averaged results are reported.

Evaluation Protocol: For each user in the training data, the adsorption algorithm generates a list of $(item, score)$ tuples as recommendations. From this list, we remove any items that are in the training set of that user as our goal is to recommend new unseen items. We then generate an ordered list of p items sorted from highest $score$ value to lowest $score$ value. We use the sorted list to compute Average Precision at p for each user and report mean of average precision at p (MAP) values over all the users.

5 Results

We report results for single-domain (our baseline with $\alpha = 1$ and $\beta,\ \gamma = 0$) and cross-domain recommendations ($\alpha,\ \beta,\ \gamma > 0$) for the co-author, conference, and reference networks in Table 1. Discussion of results follow.

5.1 Target Recommendations: Co-authors

When the target items to be recommended are co-authors, the cross-domain approach of aggregating the recommendations (WAR) attained the highest MAP score (see Table 1 Part I) compared to aggregation of neighborhood (WAN) and the baseline. However, the best MAP scores are obtained when the value of α is close to 1 and the values for β and γ are close to 0. In other words, for the task of recommending co-authors, too much knowledge transfer from conference and reference networks might actually degrade the performance. This is evident from the decrease in the MAP scores for both WAR and WAN approaches as the values for β and γ increase in Table 1 Part I. Also, the percentage increase between the highest MAP score from the cross-domain approaches and the baseline is just 7% which indicates that the source domains do not contribute much for this task. This can be explained by the way in which links are formed in co-author networks. Authors, in general, collaborate with acquaintances as opposed to unknown authors and new collaborations based on common conferences or co-cited publications are rare, although not impossible.

5.2 Target Recommendations: Conferences

When the target items to be recommended are conferences, similar to the co-author network, the cross-domain approach of aggregating recommendations (WAR) attained highest MAP score (see Table 1 Part II) compared to the other two approaches. In fact, the MAP scores from both WAN and WAR for all values of (α, β, γ) considered are better than the baseline and the percentage increase between the highest MAP score from the cross-domain approaches and the baseline is almost 60% percent. This suggests that for the task of suggesting conferences to users, the knowledge from the co-author and the reference networks is very useful. This is also evident from the values of β and γ when the MAP score is highest, which indicates that the co-author network and the reference network are given 25% weight when transferring knowledge. This is intuitive because in real-world, for an author, the conferences where his co-authors frequently published and the publication venues of his frequent references are ideal conferences for him to publish.

5.3 Target Recommendations: References

Finally, when recommending references to authors, knowledge from multiple networks helped in improving the recommendation accuracy in the reference network. This result is consistent with the results observed in co-author and conference networks and can be seen in Table 1 Part III. For this task, weighted aggregation of neighborhood (WAN) attained the highest MAP score compared to WAR and baseline approaches and the percentage increase between the highest MAP score from the cross-domain approaches and the baseline is approximately 14%. Collectively, this suggests that for the task of recommending interesting references to an author, knowledge about his preferences from the co-author and the conference networks help.

5.4 Discussion and Limitations

Our results indicate that when making cross-domain recommendations, knowledge about user preferences in closely related domains with same users but different items helped in increasing recommendation accuracy. This result is consistent for all three domains considered. However, when transferring knowledge from multiple source domains, it is important to understand that the source domains and the target domain do not and should not contribute equally to the recommendation problem, given the differences in items and preference patterns across domains. For example, for recommending co-authors, knowledge from the other two domains is not very helpful, whereas the knowledge from the co-author and the reference domains helped greatly to improve the accuracy in the conference domain. Hence, in this work, we used a weighted aggregation approach and varied the values for parameters $\alpha,\ \beta,\ \gamma$ to control the amount of knowledge used from each domain.

While the proposed approaches effectively combine user preferences from multiple domains, they also suffer from some limitations. In particular, our analysis indicates that the key to success of cross-domain recommendation is the choice of weights used to transfer knowledge between

Table 1: The target domains used in this work are shown in the first column. In second column, we show the proposed cross-domain approaches and the MAP values for these approaches for different α, β, γ combinations are shown in the next set of columns. Finally, we show the MAP score for baseline in the last column. The neighborhood size (k) and number of recommendations (p) is set to 5 and 10, respectively. The results are averaged over 5 splits of 50% of training data, selected randomly and the highest MAP value(s) for a domain is/are highlighted in bold.

Target Network	Cross-domain Algorithm	Knowledge Transfer Parameters					Baseline
		$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1$
		$\beta = 0.25$	$\beta = 0.2$	$\beta = 0.15$	$\beta = 0.1$	$\beta = 0.05$	$\beta = 0$
		$\gamma = 0.25$	$\gamma = 0.2$	$\gamma = 0.15$	$\gamma = 0.1$	$\gamma = 0.05$	$\gamma = 0.0$
I. Co-Author	WAN	0.0019	0.0022	0.0028	0.0038	0.0054	0.0070
	WAR	0.0070	0.0073	0.0074	**0.0075**	**0.0075**	
II. Conference	WAN	0.0318	0.0296	0.0284	0.0281	0.0281	0.0261
	WAR	**0.0481**	0.0476	0.0469	0.0462	0.0452	
III. Reference	WAN	0.0090	0.0116	**0.0139**	0.0133	0.0133	0.0122
	WAR	0.0130	0.0133	0.0130	0.0128	0.0127	

source and target domains. However, accurately determining the closeness between the different domains to identify the best weights to use is still a challenging research problem.

6 Conclusions and Future Work

In this paper, we identified several assumptions from existing research for cross-domain recommender systems that do not hold for some datasets of interest to us, and proposed two ways to combine user preferences from multiple domains. We conducted a study on a subset of the *DBLP* citation network and our evaluation revealed that the knowledge about co-authors, conferences, and references can be collectively used to improve recommendation accuracy for each task. Our analysis also suggested that the amount of information transferred from different domains should be carefully controlled to avoid performance decrease. For future works, we will study ways to identify better weights for transfer of knowledge between different domains. We also aim to extend this work to other heterogeneous settings with *implicit* feedback data, e.g., *LinkedIn*.

References

Adomavicius, G., and Tuzhilin, A. 2011. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.* 17(6).

Baluja, S.; Seth, R.; Sivakumar, D.; Jing, Y.; Yagnik, J.; Kumar, S.; Ravichandran, D.; and Aly, M. 2006. Video suggestion and discovery for youtube: Taking random walks through the view graph. In *Proc. of WWW*.

Bell, R. M., and Koren, Y. 2007. Improved neighborhood-based collaborative filtering. In *1st KDDCup'07*.

Cremonesi, P.; Koren, Y.; and Turrin, R. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proc. of RecSys*.

Desrosiers, C., and Karypis, G. 2011. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*. Springer.

Gao, S.; Luo, H.; Chen, D.; Li, S.; Gallinari, P.; and Guo, J. 2008. Cross-domain recommendation via cluster-level latent factor model. In *ECML/PKDD*.

Herlocker, J. L.; Konstan, J. A.; and Riedl, J. 2000. Explaining collaborative filtering recommendations. In *Proc. of CSCW*.

Li, B.; Yang, Q.; and Xue, X. 2009a. Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In *Proc. of IJCAI*.

Li, B.; Yang, Q.; and Xue, X. 2009b. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proc. of ICML*.

Pan, W.; Xiang, E. W.; Liu, N. N.; and Yang, Q. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Proc. of AAAI*.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW*.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proc. of SIGKDD*.

Talukdar, P. P., and Crammer, K. 2009. New regularized algorithms for transductive learning. In *Proc. of ECML PKDD*.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, 990–998.

Winoto, P., and Tang, T. 2008. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing* 26(3).