Cross-Domain Recommendations with Overlapping Items

Denis Kotkov, Shuaiqiang Wang and Jari Veijalainen
University of Jyvaskyla, Dept. of Computer Science and Information Systems,
P.O.Box 35, FI-40014 Jyvaskyla, Finland

Keywords: Recommender Systems, Cross-Domain Recommendations, Collaborative Filtering, Content-Based Filtering, Data Collection.

Abstract: In recent years, there has been an increasing interest in cross-domain recommender systems. However, most existing works focus on the situation when only users or users and items overlap in different domains. In this paper, we investigate whether the source domain can boost the recommendation performance in the target domain when *only items overlap*. Due to the lack of publicly available datasets, we collect a dataset from two domains related to music, involving both the users' rating scores and the description of the items. We then conduct experiments using collaborative filtering and content-based filtering approaches for validation purpose. According to our experimental results, the source domain can improve the recommendation performance in the target domain when only items overlap. However, the improvement decreases with the growth of non-overlapping items in different domains.

1 INTRODUCTION

Recommender systems use past user behavior to suggest items interesting to users (Ricci et al., 2011). An item is a piece of information that refers to a tangible or digital object, such as a good, a service or a process that a recommender system suggests to the user in an interaction through the Web, email or text message.

The majority of recommender systems suggest items based on a single domain. In this paper, the term *domain* refers to "a set of items that share certain characteristics that are exploited by a particular recommender system" (Fernández-Tobías et al., 2012). These characteristics are items' attributes and users' ratings.

However, the single domain recommender systems often suffer from data sparsity and cold start problems. In order to overcome these limitations it is possible to consider data from different domains. Recommender systems that take advantage of multiple domains are called *cross-domain recommender systems* (Fernández-Tobías et al., 2012; Cantador and Cremonesi, 2014).

In this paper, we consider a *cross-domain recommendation task* (Cantador and Cremonesi, 2014), that requires one *target domain* and at least one *source domain*. The former refers to the domain that suggested items are picked from, and similarly the latter refers to the additional domain that contains auxiliary information.

Cross-domain recommender systems can be classified based on domain levels (Cantador and Cremonesi, 2014):

- attribute level - items can be assigned to different domains according to their descriptions. One may contain jazz music, while another may consist of pop audio recordings;
- type level - items may have different types, but share common attributes. Movie and book domains have common genres, such as drama, comedy and horror, while movies and books have different types;
- item level - items from different domains may have completely different attributes and types. Songs and books might not share any common attributes;
- system level - items may belong to different recommender systems, have the same type and share many common attributes. For example, movies from IMDb[1] and MovieLens[2] may belong to different domains.

Depending on whether overlapping occurs in the set of users or items (Cremonesi et al., 2011), there

[1] http://www.imdb.com/
[2] https://movielens.org/

Kotkov, D., Wang, S. and Veijalainen, J.
Cross-Domain Recommendations with Overlapping Items.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 131-138
ISBN: 978-989-758-186-1

are four situations that enable cross-domain recommendations: a) no overlap between items and users, b) user sets of different domains overlap, c) item sets overlap, and d) item and user sets overlap.

In this work, we investigate whether the source domain improves the recommendation performance in the target domain on system level in the situation when only items overlap. The idea behind the paper is as follows. Traditional cross-domain recommender systems utilize overlapping users to discover additional interests of users, leading to the improvement of the recommendation diversity. When the items overlap, the source domain lets detect more accurate similarities between items, which should positively result in recommendation performance in the target domain.

Due to the lack of publicly available datasets for cross-domain recommender systems with overlapping items (Berkovsky et al., 2008; Kille et al., 2013), we collected data from Vkontakte[3] (VK) – Russian online social network (OSN) and Last.fm[4] (FM) – music recommender service. We then matched VK and FM audio recordings and developed the cross-domain recommender system that suggests VK recordings to VK users based on data from both domains. Each audio recording is represented by its metadata excluding the actual audio file. VK recordings thus represent the target domain, while the source domain consists of FM recordings. VK and FM recordings share titles and artists, but have different user ratings and other attributes.

In order to address the research question and illustrate the potential of additional data, we chose simple but popular recommendation algorithms to conduct experiments for validation: collaborative filtering based on users' ratings and content-based filtering based on the descriptions of the items.

Our results indicate that the source domain can improve the recommendation performance in the target domain. Furthermore, with the growth of non-overlapping items in different domains, the improvement of recommendation performance decreases. This paper thus has the following contributions:

- we initially investigate the cross-domain recommendation problem in the situation when only items overlap;
- we collect a novel dataset to conduct the experiments for addressing the research question.

The paper might be useful in real life scenarios. For example, according to our results, the performance of a recommender system lacking user ratings to achieve an acceptable performance can be improved using ratings collected from another recommender system that suggests items of the same type. However, the performance might decrease if the recommender systems do not have enough overlapping items.

The rest of the paper is organized as follows. Section 2 overviews related works. Section 3 describes the datasets used to conduct experiments. Section 4 is dedicated to recommendation approaches, while section 5 describes conducted experiments. Finally, section 6 draws final concussions.

[3]http://vk.com/

[4]http://last.fm/

2 RELATED WORKS

Most existing approaches consider additional information about users to boost the recommendation performance. For example, one of the first studies dedicated to cross-domain recommender systems investigated the effectiveness of source domains with overlapping users (Winoto and Tang, 2008). In the experiment, undergraduates from a local university were asked to rate items from different domains, such as movies, songs and books. The authors measured recommendation performance in different domain combinations and concluded that source domains decrease the recommendation performance, but may improve the diversity of recommendations.

In contrast, other studies demonstrated that source domains can boost the recommendation performance in the target domain in situations when users or both users and items overlap. For example, Sang demonstrated the feasibility of utilizing the source domain. The study was conducted on a dataset collected from Twitter[5] and YouTube[6]. The author established relationships between items from different domains using topics (Sang, 2014). Similarly to Sang, Shapira et al. also linked items from different domains, where 95 participants rated movies and allowed the researches to collect data from their Facebook pages. The results suggested that additional domains improve the recommendation performance (Shapira et al., 2013). Another study with positive results was conducted by Abel et al. The dataset contained information related to the same users from 7 different OSNs (Abel et al., 2013). Sahebi et al. demonstrated the usefulness of recommendations based on additional domains to overcome cold start problem (Sahebi and Brusilovsky, 2013).

Most works on cross-domain recommender systems focus on the situation when users or both users

[5]https://twitter.com/

[6]https://www.youtube.com/

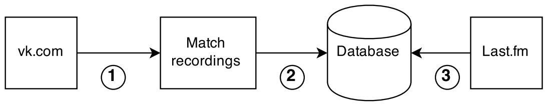


Figure 1: Data collection chart.

and items of several domains overlap (Cantador and Cremonesi, 2014). However, to the best of our knowledge, the efforts on the impact of source domains on the target domain with only overlapping items involving a real cross-domain dataset is very limited.

3 DATASETS

Due to the lack of publicly available datasets for cross-domain recommender systems with overlapping items (Berkovsky et al., 2008; Kille et al., 2013) we collected data from VK and FM. The construction of the dataset included three phases (Figure 1): 1) VK recordings collection, 2) duplicates matching, and 3) FM recordings collection.

3.1 VK Recordings Collection

The VK interface provides the functionality to add favorite recordings to users' pages. By generating random user ids we collected disclosed VK users' favorite audio recordings using VK API. Our VK dataset consists of 97,737 (76,177 unique) audio recordings added by 864 users.

Each VK user is allowed to share any audio or video recording. The interface of the OSN provides the functionality to add favorite recordings to the users page. VK users are allowed not only to add favorite audio recordings to their pages, but also to rename them. The dataset thus contains a noticeable number of duplicates with different names. To assess this number we randomly selected 100 VK recordings and manually split them into three categories:

- correct names - the name of the recording is correctly written without any grammatical mistakes or redundant symbols;
- misspelled names - the name is guessable, even if the name of the recording is replaced with the combination of artist and recording name or lyrics;
- meaningless names – the name does not contain any information about the recording. For example, "unknown" artist and "The song" recording.

Out of 100 randomly selected recordings we detected 14 misspelled and 2 meaningless names. The example can be seen from table 1.

Table 1: Examples of recordings.

Artist name	Recording name
Correct names	
Beyonce	Halo
Madonna	Frozen
Misspelled	
Alice DJ	Alice DJ - Better of Alone.mp3
Reamonn	Oh, tonight you kill me with your smile
● Lady Gaga	Christmas Tree
Meaningless	
Unknown	classic
Unknown	party

3.2 Duplicates Matching

In order to match misspelled recordings, we developed a duplicate matching algorithm that detects duplicates based on recordings' names, mp3 links and durations. The algorithm compares recordings' names based on Levenshtein distance and the number of common words excluding stop words.

We then removed some popular meaningless recordings such as "Unknown", "1" or "01", because they represent different recordings and do not indicate users' preferences. Furthermore, some users assign wrong popular artists' names to the recordings. To restrict the growth of this kind of mistakes, the matching algorithm considers artists of the duplicate recordings to be different. By using the presented matching approach, the number of unique recordings decreased from 76,177 to 68,699.

3.3 FM Recordings Collection

In order to utilize the source domain we collected FM recordings that correspond to 48,917 selected VK recordings that were added by at least two users or users that have testing data. Each FM recording contains descriptions such as FM tags added by FM users. FM tags indicate additional information such as genre, language or mood. Overall, we collected 10,962 overlapping FM recordings and 20,214 (2,783 unique) FM tags.

It is also possible to obtain FM users who like a certain recording (top fans). For each FM recording, we collected FM users who like at least one more FM recording from our dataset according to the distribution of VK users among those recordings. In fact, some unpopular FM recordings are missing top fans. We thus collected 17,062 FM users, where 7,083 of them like at least two recordings from our database.

In this work, we constructed three datasets. Each of them includes the collected FM data and different parts of the VK data:

- 0% - the dataset contains only overlapping recordings rated by VK and FM users;
- 50% - the dataset contains overlapping recordings and the half of randomly selected VK recordings that do not correspond to FM recordings;
- 100% - the dataset contains all collected VK and FM recordings.

The statistics of the datasets are presented in table 2. The number of VK users varies in different dataset, due to the lack of ratings after removing non-overlaping VK recordings.

4 RECOMMENDATION APPROACHES

In order to emphasize the importance of additional data we implemented simple, but popular collaborative filtering and content-based filtering algorithms.

4.1 Item-based Collaborative Filtering

Each recording is represented as a vector in the multidimensional feature space, where each feature is a user's choice. VK recording is represented as follows: $i_j^{vk} = (u_{1,j}^{vk}, u_{2,j}^{vk}, ..., u_{n,j}^{vk})$, where $u_{k,j}^{vk}$ equals to 1 if VK user k picks VK recording j and 0 otherwise. The representation changes if we consider the FM users: $i_j^{vk,fm} = (u_{1,j}^{vk}, u_{2,j}^{vk}, ..., u_{n,j}^{vk}, u_{1,j}^{fm}, u_{2,j}^{fm}, ..., u_{n,j}^{fm})$.

In order to rank items in the suggested list we use sum of similarities of recordings (Ekstrand et al., 2011):

$$score(u_k^{vk}, i_j^{vk}) = \sum_{i_h^{vk} \in I_{u_k^{vk}}} sim(i_j^{vk}, i_h^{vk}), \quad (1)$$

where $I_{u_k^{vk}}$ is the set of items picked by u_k^{vk} user. We use conditional probability as similarity measure (Ekstrand et al., 2011):

$$p(i_j, i_h) = \frac{Freq(i_j \wedge i_h)}{Freq(i_j) \cdot Freq(i_h)^{\alpha}}, \quad (2)$$

where $Freq(i_j)$ is the number of users that liked item i_j, while $Freq(i_j \wedge i_h)$ is the number of users that liked both items i_j and i_h. The parameter α is a demping factor to decrese the similarity for popular items. In our experiments $\alpha = 1$.

It is worth mentioning that item vectors based on FM users contain remarkably more dimensions than vectors based on VK users. In order to alleviate the problem we compared recordings using the following rule:

$$sim(i_j, i_h) = \begin{cases} p(i_j^{vk}, i_h^{vk}), & \exists i_j^{vk} \wedge \exists i_h^{vk} \wedge \\ & (\nexists i_j^{fm} \vee \nexists i_h^{fm}) \\ p(i_j^{fm}, i_h^{fm}), & \exists i_j^{fm} \wedge \exists i_h^{fm} \wedge \\ & (\nexists i_j^{vk} \vee \nexists i_h^{vk}) \\ p(i_j^{vk,fm}, i_h^{vk,fm}), & \exists i_j^{vk} \wedge \exists i_h^{vk} \wedge \\ & \exists i_j^{fm} \wedge \exists i_h^{fm} \end{cases}. \quad (3)$$

We compare items in each pair using only domains that contain users' ratings for both items.

4.2 Content-based Filtering

In a content-based approach similarly to an item-based approach each recording is represented as a vector, but each dimension corresponds to an attribute of the item. In our case, these attributes are VK FM artists and FM tags. It is worth mentioning that FM and VK artists correspond to each other.

An audio recording thus is represented as follows: $i_j^a = (a_{1,j}, a_{2,j}, ..., a_{d,j})$, where $a_{k,j}$ equals to 1 if the recording i_j is performed by the artist a_k and 0 otherwise. The user then can be represented similarly: $u_j^a = (a_{1,j}, a_{2,j}, ..., a_{d,j})$, where $a_{k,j}$ equals to 1 if user k picks the recording perfromed by the artist a_k and 0 otherwise.

The representation chages if we consider FM tags: $i_j^t = (w_{1,j}, w_{2,j}, ..., w_{q,j})$, where $w_{k,j}$ corresponds to the term frequencyinverse document frequency (Lops et al., 2011). The user vector then is denoted as follows: $u_j^t = (t_{1,j}, t_{2,j}, ..., t_{q,j})$, where $t_{k,j}$ is a number of recodings that have tag t_k and are picked by user u_j.

The recommender system compares audio recordings' vectors and a user vector using cosine similarity (Ekstrand et al., 2011). First, the suggested list is sorted according to the similarity based on artists. Second, list fragments that consist of items with the same artists' similarity are sorted according to the FM tag similarity.

5 EXPERIMENTS

In this section, we conduct experiments to demonstrate whether the source domain improves the recommendation performance in the target domain when only items overlap.

5.1 Evaluation Metrics

We used precision@K, recall@K, mean average precision (MAP) and normalized discounted cumulative gain (NDCG) to evaluate our approaches (Zhao,

Table 2: The Statistics of the Datasets.

	0%		50%		100%	
	VK	FM	VK	FM	VK	FM
Users	661	7,083	850	7,083	864	7,083
Ratings	14,207	40,782	62,435	40,782	96,737	40,782
Items	4,605	4,605	39,831	4,605	68,699	4,605
Artists	1,986	1,986	19,930	1,986	31,861	1,986
Tags	-	20,167	-	20,167	-	20,167

2013), as these metrics are the most popular in information retrieval. Precision@K, recall@K and mean average precision (MAP) are used to assess quality of recommended lists with binary relevance. Binary relevance requires each item to be relevant or irrelevant for a particular user. As in our case a user can only indicate relevance of a recording by adding it to her page, we regarded added recordings as equally relevant for a user. We regarded the rest recordings as irrelevant. Precision@K and recall@K for a specific user are calculated as follows:

$$P_u@K = \frac{r_u(K)}{K}, \tag{4}$$

$$R_u@K = \frac{r_u(K)}{r_u}, \tag{5}$$

where $r_u(K)$ is the number of items relevant for user u in the first K results, while r_u indicates the number of relevant items in the list recommended to user u. Overall precision@K and recall@K are average values.

$$Precision@K = \frac{1}{||U||}\sum_{u\in U} P_u@K, \tag{6}$$

$$Recall@K = \frac{1}{||U||}\sum_{u\in U} R_u@K, \tag{7}$$

where U is a set of evaluated users. MAP then can be calculated in the following way:

$$MAP = \frac{1}{||U||}\sum_{u\in U}\frac{1}{r_u}\left(\sum_{i=1}^{h} r_{u,i}\cdot P_u@i\right), \tag{8}$$

where $r_{u,i} = 1$ if an item at position i in the recommended list is relevant for user u and $r_{u,i} = 0$ otherwise. In our experiments h was set to 30.

We also evaluated our approaches using NDCG (Järvelin and Kekäläinen, 2002). The metric considers positions of items in recommended lists and multiple levels of relevance. We employed NDCG to measure the quality of recommendations with binary relevance. The metric is calculated as follows:

$$NDCG_u@K = Z_n \cdot \sum_{i=1}^{K} \begin{cases} 2^{r_{u,i}} - 1, & i = 1 \\ \frac{2^{r_{u,i}}-1}{log_2(i)}, & i > 1 \end{cases}, \tag{9}$$

$$NDCG@K = \frac{1}{||U||}\sum_{u\in U} NDCG_u@K, \tag{10}$$

where Z_n is the normalization constant.

5.2 Results

Following the datasets sampling strategy in (Ekstrand et al., 2011), we split each of our datasets into training and test datasets. In particular, we selected 40% of the users who rated the most VK recordings, and then chose 30% of their ratings as the testing dataset. We then regarded the rest ratings as the training dataset.

We used offline evaluation to compare results of proposed methods with baselines. The recommender system suggested 30 popular VK recordings to each testing VK user excluding recordings that the user has already added in the training set. In each approach the recommendation list consists of the same items. We chose popular items for evaluation, due to the high probability that users have seen them already.

In this study, we demonstrate the performance improvement resulting from the source domain with three simple but popular algorithms: (1) POP, (2) Collaborative Filtering (CF), and (3) Content-based Filtering (CBF). In particular, POP is a non-personalized recommendation algorithm, which orders items in the suggested list according to their popularity in the VK dataset. For the CF and the CBF algorithms, we obtained two performance results based on only VK and VK+FM data.

- **POP** - ordering items according to their popularity using the VK dataset.
- **CF(VK)** - item-based collaborative filtering using the VK dataset.
- **CF(VK+FM)** - item-based collaborative filtering using VK and FM datasets.
- **CBF(VK)** - content-based filtering using the VK dataset.
- **CBF(VK+FM)** - content-based filtering using VK and FM datasets.

Figures 2, 3 and 4 demonstrate the experimental results based on three datasets presented in Section 3. From the figures we can observe that:

1. The source domain can improve the recommendation performance in the target domain when only items overlap. For the 0% dataset, the CF algorithm achieves 0.0216, 0.0273, 0.0139 and 0.0287

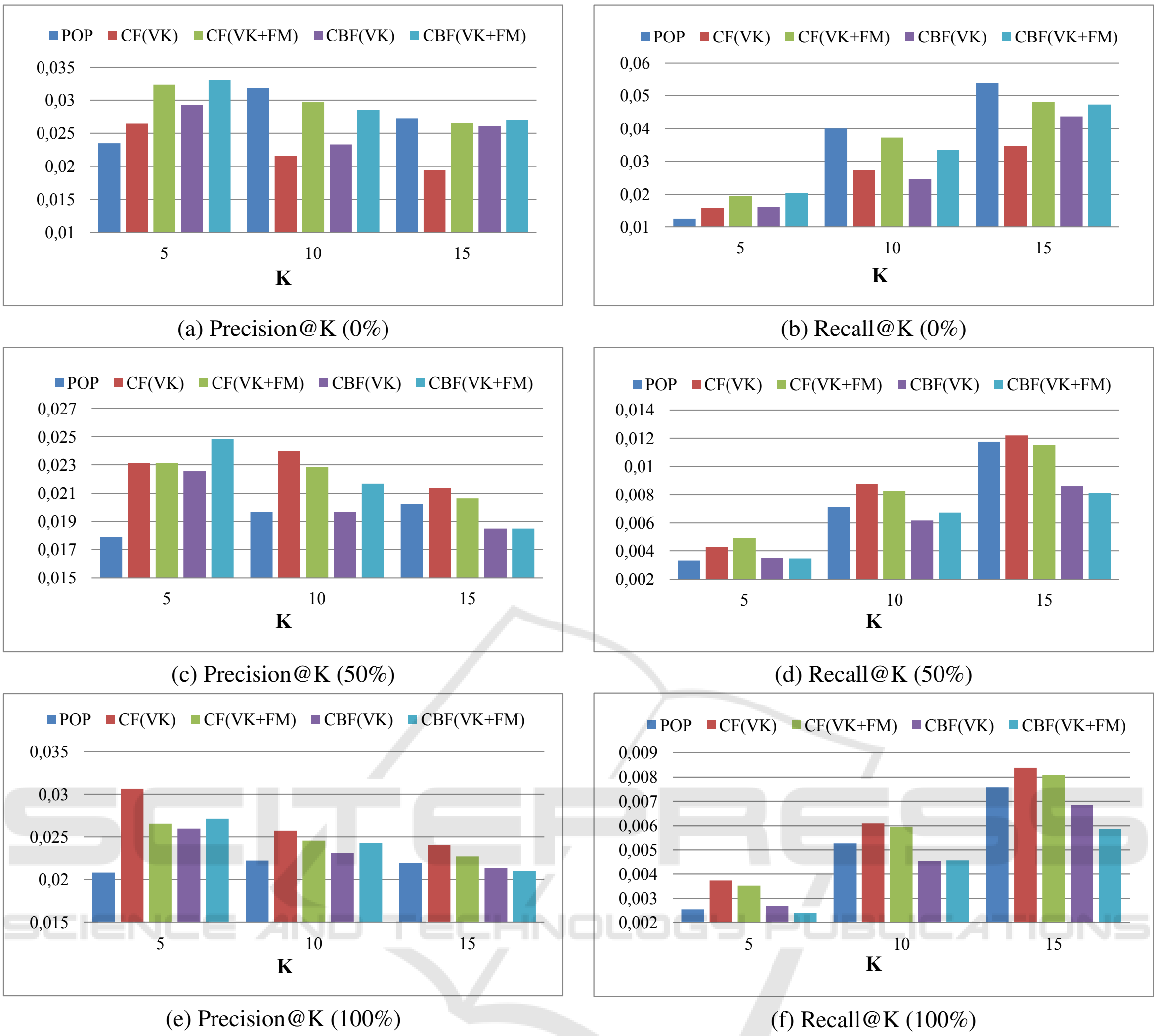


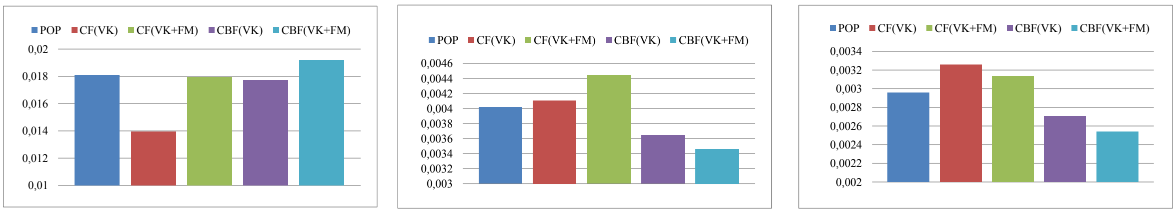
Figure 2: Precision@K and Recall@K for experiments conducted using datasets with different fractions of non-overlapping items.

in terms of precision@10, recal@10, MAP and NDCG@10 based on VK dataset, while these numbers are 0.0297, 0.0372, 0.0179 and 0.0387 based on VK+FM dataset, making the improvement of 37.5%, 36.3%, 28.8% and 34.8%, respectively. Similar improvements can be observed for the CBF algorithm.

2. The improvement declines with the growth of non-overlapping items in different domains. For example, the improvement of CBF in terms of NDCG@10 decreases as follows: 20.1%, 5.4% and 5.0% using 0%, 50% and 100% datasets, respectively. For the CF algorithm, the declining trend is even sharper. The source domain decreases the performance of the CF algorithm by 11.8% and 7.0% in terms of NDCG@5 and NDCG@10 respectively using 100% dataset. A similar trend can be observed for other numbers of first *K* results and evaluation metrics.

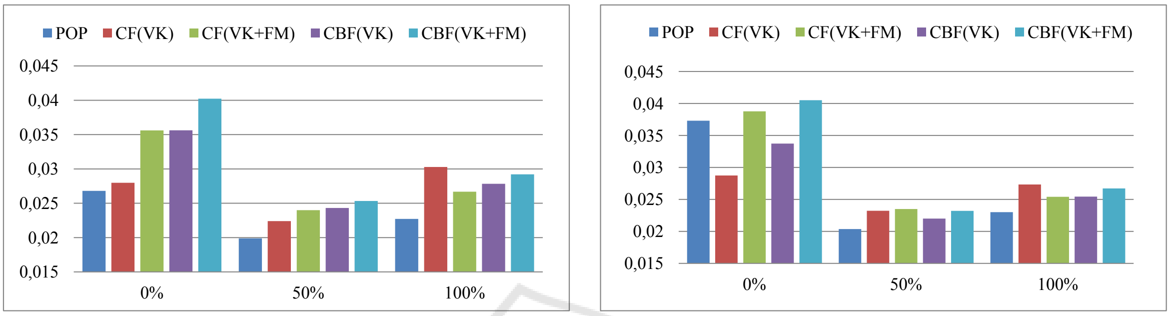
3. CF(VK) and CBF(VK) perform worse than POP in different cases, especially using the dataset that contains only overlapping recordings (0%). CF(VK) algorithm outperforms the popularity baseline with the increase of non-overlapping recordings. CF(VK) achieves 0.0139, while POP outperforms them with the number 0.0180 in terms of MAP using 0% dataset. For 100% dataset the situation is opposite. POP achieves 0.0029, while CF(VK) reaches 0.0031. POP outperforms CBF(VK) algorithm in most cases. For 0% and 100% datasets, CBF(VK) performs 1.9% and 8.4% worse than POP in terms of MAP, respectively.

Observation 1 illustrates the global correlation of



(a) MAP (0%) (b) MAP (50%) (c) MAP (100%)

Figure 3: MAP (30 recommendations) for experiments conducted using datasets with different fractions of non-overlapping items.



(a) NDCG@5 (b) NDCG@10

Figure 4: NDCG@K for experiments conducted using datasets with different fractions of non-overlapping items.

users' preferences in different domains (Winoto and Tang, 2008; Fernández-Tobías et al., 2012). Although, the data belongs to different domains, users' ratings from the source domain indicate similarities between items that improve the recommendation performance in the target domain.

Observation 2 supports the claim (Fernández-Tobías et al., 2012), that the improvement cased by the source domain rises with the growth of the overlap between target and source domains. The decrease in the recommendation performance of the CF algorithm with the FM data is caused by the different lengths of item vectors in source and target domains, where vectors of FM items contain significantly more dimensions than vectors of VK items.

In observation 3, the non-personalized algorithm POP outperforms both the personalized algorithms in different cases. CF algorithm performs worse than POP due to data sparsity, which is alleviated by adding more VK recordings to the dataset. Low performance of CBF is caused by the poor quality of item descriptions, as in the VK dataset items are described with artists only.

Figures 2, 3 and 4 demonstrate four evaluation metrics that are not always consistent. However, the described observations can still be notices.

6 CONCLUSION

In this paper we investigated cross-domain recommendations in the situation when only items overlap on system level. We collected data from VK and FM and built three datasets that contain different fractions of non-overlapping items from source and target domains. We then conducted experiments using collaborative filtering and content-based filtering algorithms to demonstrate the importance of additional data.

According to our results, the source domain can boost the recommendation performance in the target domain when only items overlap resulting from the correlation of users' preferences among different domains (Winoto and Tang, 2008). However, similarly to (Fernández-Tobías et al., 2012) our results indicated that the more items overlap in source and target domains with respect to the whole dataset the higher the improvement.

ACKNOWLEDGEMENT

The research at the University of Jyväskylä was performed in the MineSocMed project, partially supported by the Academy of Finland, grant #268078.

REFERENCES

Abel, F., Herder, E., Houben, G.-J., Henze, N., and Krause, D. (2013). Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209.

Berkovsky, S., Kuflik, T., and Ricci, F. (2008). Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286.

Cantador, I. and Cremonesi, P. (2014). Tutorial on cross-domain recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 401–402, New York, NY, USA. ACM.

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

Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173.

Fernández-Tobías, I., Cantador, I., Kaminskas, M., and Ricci, F. (2012). Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Kille, B., Hopfgartner, F., Brodt, T., and Heintz, T. (2013). The plista dataset. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, NRS '13, pages 16–23, New York, NY, USA. ACM.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer US.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to Recommender Systems Handbook*. Springer US.

Sahebi, S. and Brusilovsky, P. (2013). Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 289–295. Springer Berlin Heidelberg.

Sang, J. (2014). Cross-network social multimedia computing. In *User-centric Social Multimedia Computing*, Springer Theses, pages 81–99. Springer Berlin Heidelberg.

Shapira, B., Rokach, L., and Freilikhman, S. (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247.

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):209–225.

Zhao, Y.-L. (2013). *Community Learning in Location-based Social Networks*. Thesis. Ph.D.