This is the final draft of the following article:
Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. Improving Serendipity and Accuracy in Cross-Domain Recommender Systems. Web Information Systems and Technologies. Springer, Cham, 2017. doi: 10.1007/978-3-319-66468-2_6

Improving Serendipity and Accuracy in Cross-Domain Recommender Systems *

Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen

University of Jyvaskyla, Dept. of Computer Science and Information Systems,
P.O.Box 35, FI-40014 University of Jyvaskyla, Jyvaskyla, Finland
deigkotk@student.jyu.fi, {shuaiqiang.wang, jari.veijalainen}@jyu.fi,

Abstract. Cross-domain recommender systems use information from source domains to improve recommendations in a target domain, where the term *domain* refers to a set of items that share attributes and/or user ratings. Most works on this topic focus on accuracy but disregard other properties of recommender systems. In this paper, we attempt to improve serendipity and accuracy in the target domain with datasets from source domains. Due to the lack of publicly available datasets, we collect datasets from two domains related to music, involving user ratings and item attributes. We then conduct experiments using collaborative filtering and content-based filtering approaches for the purpose of validation. According to our results, the source domain can improve serendipity in the target domain for both approaches. The source domain decreases accuracy for content-based filtering and increases accuracy for collaborative filtering. The improvement of accuracy decreases with the growth of non-overlapping items in different domains.

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

1 Introduction

Recommender systems use past user behavior to suggest items interesting to users [17]. 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" [12]. Recommender systems use algorithms to generate recommendations.

Traditional recommendation algorithms mainly aim to improve accuracy, which indicates how good an algorithm is at suggesting items a user usually consumes. In this paper, they are referred to as *accuracy-oriented algorithms*. Generally speaking, accuracy-oriented algorithms often suggest popular items, as these items are widely consumed by individuals. To improve accuracy, recommendation algorithms also tend to suggest items similar to a user profile (a set of items rated by the user [12]), as these items match previous user tastes. As a result, a user is recommended (1) items

* This is the final draft of the paper published in the journal. The final publication is available at https://link.springer.com/chapter/10.1007/978-3-319-66468-2_6

that are popular and therefore familiar to the user [6] and (2) items that the user can easily find him/herself, which is referred to as the *overspecialization problem* [21]. In particular, as two main categories of recommendation algorithms, collaborative filtering algorithms often suggest popular items due to the popularity bias in most datasets, while content-based filtering algorithms often suffer from the overspecialization problem due to insufficient information regarding attributes of items.

Typically, the main reason why a user joins a recommender system is to find novel and interesting items the user would not find him/herself [21]. To improve user satisfaction, a recommender system should suggest serendipitous items [12]. In this paper, we follow the definitions of [10, 2, 12], which indicate that serendipitous items must be relevant, novel and unexpected to a user.

The mentioned problems can be tackled by cross-domain recommender systems, which could predict serendipitous items by enriching the training data from the target domain with additional datasets from other domains. Here the term *domain* refers to "a set of items that share certain characteristics that are exploited by a particular recommender system" [9]. These characteristics are item attributes and user ratings. Recommender systems that take advantage of multiple domains are called *cross-domain recommender systems* [9, 4, 13].

In this paper, we explore the *cross-domain recommendation task* [4, 13], that requires one *target domain* and at least one *source domain*. The former refers to the domain from which suggested items are picked from, and similarly the latter refers to the domain that contains auxiliary information.

In this work, we seek to address the following research question: Can the source domain improve serendipity in the target domain? Due to the lack of publicly available datasets for cross-domain recommender systems [3, 11, 13], we collected data from Vkontakte[1] (VK) – Russian online social network (OSN) and Last.fm[2] (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.

We regard items that share certain attributes and belong to different domains as *overlapping*, while those that do not as *non-overlapping*. In our case, VK and FM recordings that have the same titles and artists are overlapping items.

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 user ratings and content-based filtering based on the descriptions of the items.

Our results indicate that the source domain can improve serendipity in the target domain for both collaborative filtering and content-based filtering algorithms:

[1] http://vk.com/
[2] http://last.fm/

- The traditional collaborative filtering algorithms tend to suggest popular items, as most datasets contain rich information regarding these items in terms of user ratings. Combing datasets of different domains decreases the popularity bias.
- Content-based filtering algorithms often suffer from the overspecialization problem due to poor data regarding item attributes. Enriching item attributes alleviates the problem and increases serendipity.

According to our results, the source domain has a negative impact on accuracy for content-based filtering, and a positive impact on accuracy of collaborative filtering. Furthermore, with the growth of non-overlapping items in different domains, the improvement of accuracy for collaborative filtering decreases.

This paper has the following contributions:

- We initially investigate the cross-domain recommendation problem in terms of serendipity.
- We collect a novel dataset to conduct the experiments for addressing the research question.

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.

2 Related Works

In this section, we survey state-of-the-art efforts regarding serendipity and cross-domain recommendations.

2.1 Serendipity in Recommender Systems

According to the dictionary[3], serendipity is "the faculty of making fortunate discoveries by accident". The term was coined by Horace Walpole, who referenced the fairy tale, "The Three Princes of Serendip", to describe his unexpected discovery [16].

Currently, there is no agreement on definition of serendipity in recommender systems. Researchers employ different definitions in their studies. In this paper, we employ the most common definition, which indicates that serendipitous items are relevant, novel and unexpected [10, 2, 12].

Given the importance of serendipity, researchers have proposed different serendipity-oriented recommendation algorithms. For example, Lu et al. presented a serendipitous personalized ranking algorithm [15]. The algorithm is based on matrix factorization with the objective function that incorporates relevance and popularity of items. Another matrix factorization based algorithm is proposed by Zheng, Chan and Ip [24]. The authors proposed the unexpectedness-augmented utility model, which takes into account relevance, popularity and similarity of items to a user profile. In contrast, Zhang et al.

[3] http://www.thefreedictionary.com/serendipity

provided the recommendation algorithm *Full Auralist* [23]. It consists of three algorithms, each being responsible for relevance, diversity and unexpectedness. To the best of our knowledge, studies that focus on improving serendipity using source domains are of restricted availability.

2.2 Cross-Domain Recommendations

Cross-domain recommender systems use multiple domains to generate recommendations, which can be categorized based on domain levels [4, 5]:

- Attribute level. Items have the same type and attributes. Two items are assigned to different domains if they have different values of a particular attribute. A pop song and jazz song might belong to different domains.
- Type level. Items have similar types and share some common attributes. Two items are assigned to different domains if they have different subsets of attributes. A photograph and animated picture might belong to different domains. Even though both items have common attributes, such as a title, publisher and tags, other attributes might be different (duration attribute for animated pictures).
- Item level. Items have different types and all or almost all attributes. Two items are assigned to different domains if they have different types. A song and book might belong to different domains, as almost all attributes of the items are different.
- System level. Two items are assigned to different domains if they belong to different systems. For example, movies from IMDb[4] and MovieLens[5] might belong to different domains.

Depending on whether overlapping occurs in the set of users or items [7], there 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.

Most efforts on cross-domain recommendations focus on the situation when users or both users and items overlap [13]. For example, Sang demonstrated the feasibility of utilizing the source domain. The study was conducted on a dataset collected from Twitter[6] and YouTube[7]. The author established relationships between items from different domains using topics [19]. Similarly to Sang, Shapira, Rokach and Freilikhman also linked items from different domains, where 95 participants rated movies and allowed the researches to collect data from their Facebook pages [20]. The results suggested that source domains improve the recommendation performance [20]. Another study with positive results was conducted by Abel et al. The dataset contained information related to the same users from 7 different OSNs [1]. Sahebi and Brusilovsky demonstrated the usefulness of recommendations based on source domains to overcome cold start problem [18].

[4] http://www.imdb.com/
[5] https://movielens.org/
[6] https://twitter.com/
[7] https://www.youtube.com/

Most works on cross-domain recommendations focus on accuracy. To the best of our knowledge, the efforts on the impact of source domains on the target domain in terms of serendipity involving a real cross-domain dataset are very limited. In this paper, we investigate whether source domains can improve serendipity in the target domain when only items overlap on system level.

3 Datasets

Due to the lack of publicly available datasets for cross-domain recommender systems with overlapping items [3, 11] 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.

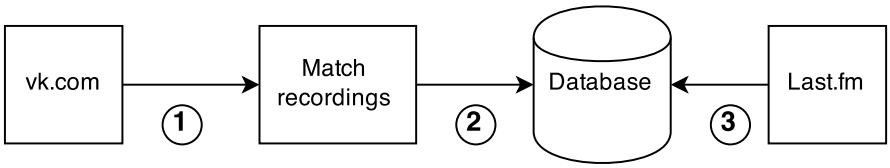


Fig. 1: Data collection chart.

3.1 VK Recordings Collection

The VK interface provides the functionality to add favored recordings to users' pages. By generating random user IDs we collected disclosed VK users' favored audio recordings using VK API. Each audio recording is represented by its metadata excluding the actual audio file. 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 favored recordings to the users page. VK users are allowed not only to add favored 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

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 the 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 user preferences. Furthermore, some users assign wrong popular artists' names to the recordings. To restrict the growth of these kinds 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

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. FM users liked $4,609$ FM recordings among those collected.

3.4 The Statistics of the Datasets

In this work, we constructed three datasets. Each of them includes the collected FM data and different parts of the VK data (percentage indicates the fraction of overlapping items):

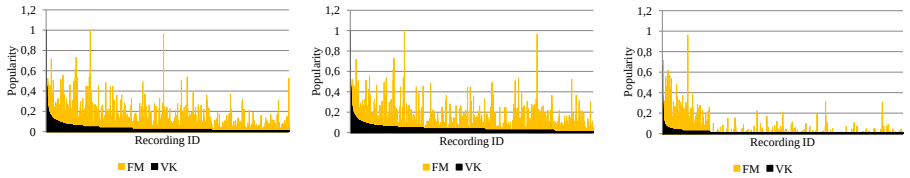
– 100% - the dataset contains only overlapping recordings picked by VK and FM users;
– 50% - the dataset contains equal number of overlapping and non-overlapping recordings;
– 7% - the dataset contains all collected VK and FM recordings. The fraction of overlapping recordings is 6.7%.

The 7% dataset contains all the collected and processed data. We presented results for 50% and 100% datasets to demonstrate how serendipity and accuracy change when a dataset contains different fraction of overlapping items.

Table 2: The Statistics of the Datasets.

	100%		50%		7%	
	VK	FM	VK	FM	VK	FM
Users	665	7,083	795	7,083	864	7,083
Ratings	14,526	40,782	33,680	40,782	96,737	40,782
Items	4,609	4,609	9,218	4,609	68,699	4,609
Artists	1,986	1,986	4,595	1,986	31,861	1,986
Tags	-	20,167	-	20,167	-	20,167

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



(a) 100% dataset (b) 50% dataset (c) 7% dataset

Fig. 2: Popularity distributions of VK and FM datasets

According to figure 2, each recording has different popularity among VK and FM users. The FM dataset contains reach information in terms of user ratings regarding recordings unpopular in the VK dataset. In the figure, popularity is based on the number of users who picked a particular item:

$$Popularity_i = \frac{Freq(i)}{Freq_{max}}, \tag{1}$$

where $Freq(i)$ is the number of users who picked recording i, while $Freq_{max}$ corresponds to the maximum number of users picked the same recording in a dataset.

4 Recommendation Approaches

In this section, we implemented simple but popular collaborative filtering and content-based filtering algorithms to demonstrate the impact of the data from source domains.

4.1 Item-Based Collaborative Filtering

We chose item-based collaborative filtering as the first experimental algorithm. It is a representative recommendation algorithm that has been widely used in industry due to its scalability [8]. In item-based collaborative filtering, each audio recording (item) is represented as a vector in a multidimensional feature space, where each feature is a user's choice (rating). VK recording is represented as follows: $i^{vk} = (u^{vk}_{1,i}, u^{vk}_{2,i}, ..., u^{vk}_{n,i})$, and each element $u^{vk}_{1,i} \in \{0,1\}$ for $k = 1,...,||U||$, where U is a set of users, while $u^{vk}_{k,i}$ equals to 1 if VK user k picks VK recording i^{vk} and 0 otherwise. To integrate the source domain (FM) with our target domain (VK), we included FM users as follows: $i^{vkfm} = (u^{vk}_{1,i}, u^{vk}_{2,i}, ..., u^{vk}_{n,i}, u^{fm}_{1,i}, u^{fm}_{2,i}, ..., u^{fm}_{n,i})$.

To generate recommendations, item-based collaborative filtering first detects recordings that are most similar to recordings picked by the target user. The algorithm then ranks recordings based on the obtained similarities.

To measure similarity, we used conditional probability, which is a common similarity measure for situations in which users only indicate items they like without specifying how much they like these items (unary data) [8]. Conditional probability is calculated as follows:

$$p(i,j) = \frac{Freq(i \wedge j)}{Freq(i) \cdot Freq(j)^{\alpha}}, \tag{2}$$

where $Freq(i)$ is the number of users that picked item i, while $Freq(i \wedge j)$ is the number of users that picked both items i and j. The parameter α is a damping factor to decrease the similarity for popular items. In our experiments $\alpha = 1$.

Item vectors based on FM users contain remarkably more dimensions than vectors based on VK users. To alleviate the problem, we compared recordings using the following rule:

$$sim(i,j) = \begin{cases} p(i^{vk}, j^{vk}), & \exists i^{vk} \wedge \exists j^{vk} \wedge \\ & (\nexists i^{fm} \vee \nexists j^{fm}) \\ p(i^{fm}, j^{fm}), & \exists i^{fm} \wedge \exists j^{fm} \wedge \\ & (\nexists i^{vk} \vee \nexists j^{vk}) \\ p(i^{vkfm}, j^{vkfm}), & \exists i^{vk} \wedge \exists j^{vk} \wedge \\ & \exists i^{fm} \wedge \exists j^{fm} \end{cases}. \tag{3}$$

We compared items in each pair using domains that contain user ratings for both items. To rank items in the suggested list, we used sum of similarities of recordings [8]:

$$score(u,i) = \sum_{j \in I_u} sim(i,j), \tag{4}$$

where I_u is the set of items picked by user u (user profile).

4.2 Content-Based Filtering

We chose content-based filtering algorithm, as this algorithm uses item attributes instead of user ratings to generate recommendations. In our case, these attributes are VK - FM artists and FM tags. Each FM artist corresponds to a particular VK artist.

To represent items, we used a common weighting scheme, term frequency-inverse document frequency (TF-IDF). TF-IDF weight consists of two parts:

$$tfidf_{attr,i} = tf_{attr,i} \cdot idf_{attr}, \tag{5}$$

where $tf_{attr,i}$ corresponds to the frequency of attribute $attr$ for item i (term frequency), while idf_{attr} corresponds to the inverse frequency of attribute $attr$ (inverse document frequency). The term frequency is based on the number of times an attribute appears among attributes of an item with respect to the number of item attributes:

$$tf_{attr,i} = \frac{n_{attr,i}}{n_i}, \tag{6}$$

where n_i is the number of attributes of item i, while $n_{attr,i}$ is the number of times attribute $attr$ appears among attributes of item i. In our case, $n_{attr,i} = 1$ for each item, while n_i varies depending on the item. The term frequency increases with the decrease of the number of item attributes. The inverse document frequency is based on the number of items with an attribute in the dataset:

$$idf_{attr} = ln\frac{||I||}{||I_{attr}||}, \tag{7}$$

where I is a set of all the items, while I_{attr} is a set of items that have attribute $attr$. The inverse document frequency is high for rare attributes and low for popular ones. TF-IDF weighting scheme assigns high weights to rare attributes that appear in items with low number of attributes.

An audio recording is represented as follows: $i^a = (a_{1,i},\ a_{2,i}, ..., a_{d,i})$, where $a_{k,i}$ corresponds to the TF-IDF weight of artist a_k [14]. The user is represented as follows: $u^a = (a_{1,u}, a_{2,u}, ..., a_{d,u})$, where $a_{k,u}$ corresponds to the number of recordings picked by user u performed by artist a_k.

To integrate FM data, we considered FM tags as follows: $i^{at} = (a_{1,i},\ a_{2,i}, ..., a_{d,i},\ t_{1,i},\ t_{2,i}, ..., t_{q,i})$, where $t_{k,i}$ corresponds to the TF-IDF weight of tag t_k [14]. The user vector then is denoted as follows: $u^{at} = (a_{1,u},\ a_{2,u}, ..., a_{d,u},\ t_{1,u}, t_{2,u}, ..., t_{q,u})$, where $t_{k,u}$ is the number of recordings picked by user u having tag t_k.

The recommender system compares audio recordings' vectors and a user vector using cosine similarity [8]:

$$cos(u,i) = \frac{u \cdot i}{||u||||i||}, \tag{8}$$

where u and i are user and item vectors. To suggest recordings, content-based filtering ranks recordings according to $cos(u,i)$. In our experiments, we used $cos(u^a, i^a)$ for VK data and $cos(u^{at}, i^{at})$ for VK and FM data.

5 Experiments

In this section, we detail experiments conducted to demonstrate whether the source domain improves serendipity and accuracy in the target domain when only items overlap.

5.1 Evaluation Metrics

To assess the performance of algorithms we used two metrics: (1) *Precision@K* to measure accuracy and (2) a traditional serendipity metric *Ser@K*.

Precision@K is a commonly used metric to assess quality of recommended lists with binary relevance. In our datasets, recordings added by a user to his/her page are relevant, while the rest of the recordings are irrelevant to the user. *Precision@K* reflects the fraction of relevant recordings retrieved by a recommender system in the first K results. The metric is calculated as follows:

$$Precision@K = \frac{1}{||U||} \sum_{u \in U} \frac{||RS_u(K) \cap REL_u||}{K}, \tag{9}$$

where U is a set of users, while $RS_u(K)$ is a set of top-K suggestions for user u. Recordings from the test set (ground truth) for user u are represented by REL_u.

The traditional serendipity metric is based on (1) a primitive recommender system, which suggests items known and expected by a user, and (2) a set of items similar to a user profile. Evaluated recommendation algorithms are penalized for suggesting items that are irrelevant, generated by a primitive recommender system or included in the set of items similar to a user profile. Similarly to [2], we used a slight modification of the serendipity metric:

$$Ser@K = \frac{1}{||U||} \sum_{u \in U} \frac{||(RS_u(K) \backslash PM \backslash E_u) \cap REL_u||}{K}, \tag{10}$$

where PM is a set of suggestions generated by the primitive recommender system, while E_u is a set of recordings similar to recordings picked by user u. We selected the 10 most popular recordings for PM following one of the most common strategies [24, 15]. Set of items similar to a user profile E_u represents all the recordings that have common artists with recordings user u picked. User u can easily find recordings from set E_u by artist name, we therefore regard these recordings as obvious.

5.2 Results

Following the datasets sampling strategy in [8], we split each of our datasets into training and test datasets and applied 3-fold cross-validation. We selected 40% of the users who picked the most VK recordings, and chose 30% of their ratings as the testing dataset. We then regarded the rest of the ratings as the training dataset.

To compare the results of various baselines, we used offline evaluation. The recommender system suggested 30 popular VK recordings to each testing VK user excluding

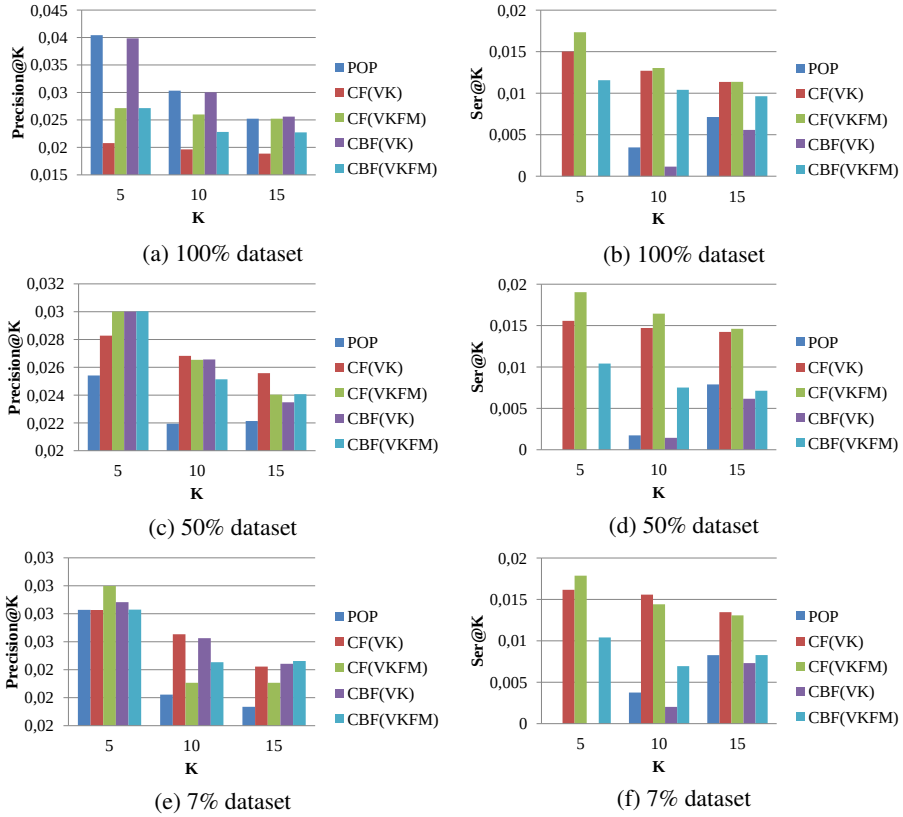
recordings that the user has already added in the training set. In each approach the recommendation list consists of the same items. We chose popular items for evaluation, as the users are likely to be familiar with those items.

In this study, we demonstrate serendipity and accuracy improvements resulting from the source domain with three simple but popular algorithms: (1) POP, (2) Collaborative Filtering (CF), and (3) Content-Based Filtering (CBF). It is important to note that 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 (1) data collected from VK and (2) data collected from both VK and FM.

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

Figure 3 demonstrates the experimental results based on three datasets presented in section 3. From the figure we can observe that:

1. The source domain can improve serendipity in the target domain. On all datasets, CBF based on VK and FM data outperforms CBF based on only VK data in terms of serendipity. For collaborative filtering the situation is very similar, except the decrease of serendipity for recommendation lists of length 10 and 15 on the 7% dataset. For the 50% dataset, the CF algorithm achieves 0.0156, 0.0147 and 0.0142 in terms of $Ser@5$, $Ser@10$ and $Ser@15$ based on VK data, while these numbers are 0.0190, 0.0164 and 0.0146 based on VK and FM data, making the improvement of 22.2%, 11.7% and 2.7%, respectively.
2. For collaborative filtering, the source domain can improve accuracy in the target domain when only items overlap. For the 100% dataset, the CF algorithm achieves 0.0208, 0.0196 and 0.0189 in terms of $Precision@5$, $Precision@10$ and $Precision@15$ based on VK data, while these numbers are 0.0271, 0.0260 and 0.0252 based on VK and FM data, making the improvement of 30.6%, 32.4% and 33.7%, respectively.
3. The improvement of accuracy declines with the growth of non-overlapping items for collaborative filtering. The improvement of CF in terms of $Precision@5$ decreases as follows: 30.6%, 6.1% and 6.0% using 100%, 50% and 7% datasets, respectively.
4. The source domain decreases accuracy of content-based filtering. For the 100% dataset, CBF based on VK and FM data decreases $Precision@5$, $Precision@10$ and $Precision@15$ by 31.9%, 24.0% and 11.2%, respectively.
5. Despite being accurate, popularity baseline has a very low serendipity. POP outperforms other algorithms in terms of accuracy on the 100% dataset. Meanwhile, the algorithm fails to suggest any serendipitous items in top-5 recommendations on each dataset.



(a) 100% dataset (b) 100% dataset

(c) 50% dataset (d) 50% dataset

(e) 7% dataset (f) 7% dataset

Fig. 3: *Precision@K* and *Ser@K* for experiments conducted using datasets with different fractions of non-overlapping items.

According to observations 1 and 2, CF(VKFM) outperforms CF(VK) in terms of both serendipity and accuracy. The improvement of accuracy illustrates the global correlation of user preferences in different domains [22, 9]. Although, the data belongs to different domains, user ratings from the source domain indicate similarities between items that improve the recommendation performance in the target domain. The improvement of serendipity is caused by the growth of accuracy and by different popularity distributions in VK and FM datasets.

Observation 3 supports the claim [9], that the improvement cased by the source domain rises with the growth of the overlap between target and source domains. The decrease of accuracy for 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.

Observations 1 and 4 indicate that the FM data positively contributes to serendipity and negatively affects accuracy of the content-based filtering algorithm. As users tend to add recording of the same artist, CBF(VK) significantly outperforms CBF(VKFM). However, most recordings suggested by CBF(VK) are obvious to a user, as the user can find these recordings him/herself. As a result, the serendipity of CBF(VK) is very low. FM tags help recommend similar recordings of artists novel to the user. Recordings that share the same FM tags do not necessarily share the same artists, which results in the decrease of accuracy and increase of serendipity.

Observation 5 indicates that POP has very low serendipity, despite being accurate. Popular recommendations are likely to be accurate, as users tend to add familiar recordings. However, popular recordings are widely recognized by users and therefore regarded as obvious.

6 Conclusion

In this paper, we first initially investigated the cross-domain recommendation problem in terms of serendipity. 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 extensive experiments with collaborative filtering and content-based filtering algorithms to demonstrate the impact of source domains on performance gains of the target domain.

According to our results, the source domain can improve serendipity in the target domain when only items overlap on system level for both collaborative filtering and content-based filtering algorithms. The integration of the source domain resulted in the decrease of accuracy for content-based filtering and the increase of accuracy for collaborative filtering. Similarly to [9] our results indicated that the more items overlap in source and target domains with respect to the whole dataset the higher the improvement of accuracy for collaborative filtering.

7 Acknowledgement

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

References

1. Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. User Modeling and User-Adapted Interaction 23, 169–209 (2013)
2. Adamopoulos, P., Tuzhilin, A.: On unexpectedness in recommender systems: Or how to better expect the unexpected. ACM Transactions on Intelligent Systems and Technology 5, 1–32 (2014)
3. Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of user models for enhanced personalization in recommender systems. User Modeling and User-Adapted Interaction 18, 245–286 (2008)
4. Cantador, I., Cremonesi, P.: Tutorial on cross-domain recommender systems. In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 401–402. New York, NY, USA (2014)
5. Cantador, I., Fernández-Tobías, I., Berkovsky, S., Cremonesi, P.: Cross-domain recommender systems, pp. 919–959. Springer, Boston, MA (2015)
6. Celma Herrada, Ò.: Music recommendation and discovery in the long tail. Ph.D. thesis, Universitat Pompeu Fabra (2009)
7. Cremonesi, P., Tripodi, A., Turrin, R.: Cross-domain recommender systems. In: 11th IEEE International Conference on Data Mining Workshops. pp. 496–503 (2011)
8. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction 4, 81–173 (2011)
9. Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.: Cross-domain recommender systems: A survey of the state of the art. In: Proceedings of the 2nd Spanish Conference on Information Retrieval. pp. 187–198 (2012)
10. Iaquinta, L., Semeraro, G., de Gemmis, M., Lops, P., Molino, P.: Can a recommender system induce serendipitous encounters? IN-TECH (2010)
11. Kille, B., Hopfgartner, F., Brodt, T., Heintz, T.: The plista dataset. In: Proceedings of the 2013 International News Recommender Systems Workshop and Challenge. pp. 16–23. ACM, New York, NY, USA (2013)
12. Kotkov, D., Veijalainen, J., Wang, S.: Challenges of serendipity in recommender systems. In: Proceedings of the 12th International conference on web information systems and technologies. SCITEPRESS (2016)
13. Kotkov, D., Wang, S., Veijalainen, J.: Cross-domain recommendations with overlapping items. In: Proceedings of the 12th International Conference on Web Information Systems and Technologies. vol. 2, pp. 131–138. SCITEPRESS (2016)
14. Lops, P., de Gemmis, M., Semeraro, G.: Recommender Systems Handbook, chap. Content-based Recommender Systems: State of the Art and Trends, pp. 73–105. Springer US, Boston, MA (2011)
15. Lu, Q., Chen, T., Zhang, W., Yang, D., Yu, Y.: Serendipitous personalized ranking for top-n recommendation. In: Proceedings of the The IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology. pp. 258–265. IEEE Computer Society, Washington, DC, USA (2012)

16. Remer, T.G.: Serendipity and the three princes: From the Peregrinaggio of 1557, p. 20. University of Oklahoma Press (1965)
17. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems Handbook, chap. Introduction to Recommender Systems Handbook, pp. 1–35. Springer US (2011)
18. 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: User Modeling, Adaptation, and Personalization, pp. 289–295. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2013)
19. Sang, J.: Cross-network social multimedia computing. In: User-centric Social Multimedia Computing, pp. 81–99. Springer Theses, Springer Berlin Heidelberg (2014)
20. Shapira, B., Rokach, L., Freilikhman, S.: Facebook single and cross domain data for recommendation systems. User Modeling and User-Adapted Interaction 23, 211–247 (2013)
21. Tacchini, E.: Serendipitous mentorship in music recommender systems. Ph.D. thesis, Università degli Studi di Milano (2012)
22. 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 Generation Computing 26, 209–225 (2008)
23. Zhang, Y.C., Séaghdha, D.O., Quercia, D., Jambor, T.: Auralist: Introducing serendipity into music recommendation. In: Proceedings of the 5th ACM International Conference on Web Search and Data Mining. pp. 13–22. ACM, New York, NY, USA (2012)
24. Zheng, Q., Chan, C.K., Ip, H.H.: An unexpectedness-augmented utility model for making serendipitous recommendation. In: Advances in Data Mining: Applications and Theoretical Aspects, vol. 9165, pp. 216–230. Springer International Publishing (2015)