Utilizing Facebook Single and Cross Domain Data for Recommendation Systems

Bracha Shapira[1], Lior Rokach[1], Shirley Freilichman[2]

[1]Department of Information Systems Engineering and Deutsche-Telekom Laboratories at Ben-Gurion University,
[2]Department of Industrial Engineering and Management,
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Bshapira@bgu.ac.il, liorrk@bgu.ac.il, shirly.freilikhman@gmail.com

Abstract

The emergence of social networks and the vast amount of data that they contain about their users make them a valuable source for personal information about users for recommender systems. In this paper we investigate the feasibility and effectiveness of utilizing existing available data from social networks for the recommendation process, specifically from Facebook. The data may replace or enrich explicit user ratings. We extract from Facebook content published by users on their personal pages about their favorite items and preferences in the domain of recommendation, and data about preferences related to other domains to allow cross- domain recommendation. We study several methods for integrating Facebook data with the recommendation process and compare the performance of these methods with that of traditional collaborative filtering that utilizes user ratings. In a field study that we conducted, recommendations obtained using Facebook data were tested and compared for 95 subjects and their crawled Facebook friends. Encouraging results show that when data is sparse or not available for a new user, recommendation results relying solely on Facebook data are at least equally as accurate as results obtained from user ratings. The experimental study also indicates that enriching sparse rating data by adding Facebook data can significantly improve results. Moreover, our findings highlight the benefits of utilizing cross domain Facebook data to achieve improvement in recommendation performance.

Keywords
Recommender systems, Social Networks, Collaborative Filtering, Cross-Domain, Evaluation

This paper or a similar version is not currently under review by a journal or conference, nor will it be submitted to such within the next three months. This paper is void of plagiarism or self-plagiarism as defined in Section 1 of ACM's Policy and Procedures on Plagiarism."

1. Introduction

Recommender systems typically aim at predicting the rating or rank of relevancy of items that a user has not seen and recommend items best-suited to the active user. Recommender systems that are based on collaborative filtering (CF), which is the most commonly used technique for recommender systems, (Ricci et al., 2011), recommend items to the user that were considered relevant by other similar users (Adomavicius and Tuzhilin, 2005). Similarity between users is typically measured by common items they rated. Hence, the accuracy of collaborative filtering-based methods heavily depends on user willingness to rate items. However, since providing ratings while using a system interferes the users' regular activities and requires extra effort from them, users tend not to provide ratings. In addition, in many systems there are much more items than users that usually rate only a very small portion of the items.

The result is the well-known sparsity problem that many collaborative filtering (CF) based recommender systems experience. Sparsity is defined as the ratio between the actual ratings and the potential ratings and common sparsity values in commercial applications (e.g. Netflix) reach more than 95%. Although many studies suggest different methods to overcome or reduce data sparseness (e.g., Yildirim and Krishnamoorthy ,2008; Huang and Gong 2008, Lekakos and Giaglis, 2007) it is still a major challenge for CF- based recommender systems. Another, known challenge for recommender systems is the cold-start problem (Resnick & Varian 1997, Ricci et al., 2011). Most CF-based systems cannot provide personalized recommendations until a user performs some activities that can be used to generate her initial profile.

One method to overcome sparsity and cold start is to obtain information from alternative external sources that can substitute or complement missing rating data to enable accurate recommendations. Due to the popularity of social networks and the vast amount of personal data they hold, social networks could be used as valuable external sources. Indeed, recent numerous studies (e.g., Groh et. Al., 2007; Guy et al. 2009, Said et al., 2010) have shown that integrating data from social networks to the recommendation process contributes to the quality of recommendations in certain situations. In this paper we focus on the Facebook social network that is currently the most popular social network. At the year 2012, Facebook has almost 850 million

registered users and keeps growing[1]. We present empirical evidence to the effectiveness of obtaining and utilizing information that exists in Facebook. Specifically, we focused on the potential of utilizing relevant content from the user Facebook profile that is related to the recommended domain as well as data from other domains that are available on the profile, to allow cross domain recommendation. All this information can be implicitly inferred and available for the system without interfering the users' activities or depending on their good will.

For a remedy to the cold start problem a system may collect information about a new user in a domain from her Facebook page and generate an initial user profile by extracting data relevant to the specific domain (if such information exists), or data related to other domains, as well as utilizing information about the behavior and preferences of the new user's friends.

We present results from a user study that compared the quality of recommendations obtained when using traditional rating information with results when using Facebook data. The findings have important implications on recommender systems practice due to the availability and rise of Facebook and other alike social networks that contain such (public) data on one hand, and the difficulties of recommender systems to obtain users explicit feedback on the other hand. Our findings show that on some situations, information obtained from Facebook can substitute explicit rating and achieve not only competitive results to parallel explicit ratings but rather be superior. This result might intuitively be explained by the fact that the data that the user chooses to publish on her personal pages is more indicative to her interest then feedback that she provides to items that a system chooses for her.

We implement and apply various prediction algorithms that utilize available data from Facebook. Results are compared with two baseline CF algorithms (enhanced CF k-NN and SVD) that are based on explicit rating data that was collected specifically for this research from the same Facebook users. We also explore the effect of combining both types of data (Facebook and ratings) when such data is available.

[1] www.facebook.com/press/info.php?statistics

It is important to highlight the differences between Facebook data and rating data. Rating data is explicitly provided by the user as feedback for recommendations while Facebook data is derived from the user's published page. The explicit ratings present the user's selection from among the items that were recommended to her; Facebook reveals the user's favorite items from the items she is familiar with and decides to publish in her personal page. Intuitively this seems to be a stronger indication of user preferences than ratings that are a response to items that a system selects for the user. As explained in Yifan et al. (2008), ratings are usually defined in a specific range (e.g., 1-5) where the lowest rank indicating that the user does not like the item and the highest rank favoring of the item. Facebook data is unary and the user just indicates her interest in an item by mentioning it on her page, (there is no explicit numeric indication of high, low or negative favoring of items). Ratings are indeed more specific than an unary indication of user's preferences, but besides the burden they put on the users, rating are known to be noisy (Amatrian et al., 2009) since users are not consistent in providing their ratings. In addition, recent studies (e.g., Koren & Sill 2011) argue that numerical ratings may not reflect the user intentions well. Different users tend to have different internal scales with different distances between the scales, however, when analyzed the differences are— considered equally. .
The unary characteristic of Facebook data might put constraints on the accuracy of the prediction algorithm since the input is less specific and requires some analysis before it can be used due to the free style of publishing information on Facebook. However, given the non-interfering implicit feedback acquisition method, on one hand, and the problematic of the ratings on the other hand it seems reasonable to consider substitution or enrichment of the ratings with Facebook data when applicable. Our results indeed support this intuition.

Facebook data is also broader in scope than explicit ratings within a recommender system, since it relates to several domains (e.g., music, TV shows, actors, etc.) and might thus provide better indications for the user profile than the common single domain rating data (e.g., Netflix). In this paper we explore the effect of these properties (along with other specific properties of the current data) on recommendations results.

Recent studies, which extract data from social networks, have focused mainly on the user's social ties in attempt to compute similarity between users (e.g., Victor et al., 2011, Massa & Avessani 2007, Ma et al., 2011, Guy et al., 2009). We, however, empirically explore the Facebook network and focus not on extracting similarity between users based on their social ties, but rather on extracting users' published content (preferences) from their Facebook accounts. The extraction is performed for content related to the domain of recommendation and for other relevant domains and thus demonstrates also the effectiveness of cross-domain recommendation. We show that it is possible to avoid completely the need of collecting ratings from users if Facebook data is extracted. We experiment the effect of such a replacement on the quality of recommendations by a unique user study that acquired both, rating data and Faceook data for the same users. In addition, our study is the first to utilize and empirically examine cross-domain information from a social network (and especially Facebook) for the purpose of recommending a specific domain. Recent studies examined the effect of cross-domain information on recommendation results (Bekovsky at al., 2007, 2008; Yang and Xu 2009b), but most of them were not tested on real world cross-domain data with overlapping users but rather on data that was artificially divided into domains. Accordingly, we were able to test our cross-domain methods on different domain pairs in order to investigate the effect of domain and dataset features on results.

The contributions of this study are twofold:

- We show that Facebook data can replace or enrich explicit ratings for obtaining at least comparable recommendation results.

- We empirically define the scenarios and conditions on which cross domain data derived from Facebook personal pages can be beneficial for recommending items for one domain.

The rest of the paper is organized as follows: Section 2 describes related studies; section 3 de scribes in detail the research questions examined and section 4 presents the methods we developed to utilize Facebook data for recommendations. Section 5 details our experiments, results and discusses their significance, while section 6

describes additional experiments exploring cross domain issues. Finally section 7 concludes and highlights limitations and further studies.

2. Related work

2.1 Extraction and exploitation of users' data from social networks

A recent trend in the recommendation system research is to examine methods for integrating social related information from social networks into the recommendation process aiming at improving recommendation results. Many of these studies (e.g., Victor et al., 2011., Yuan et al., 2010, Massa & Avensi 2007, Golbeck & Hendler, 2006, Guy at al., 2007, Bourke at al., 2011, Groh et al., 2012, Spertus et al., 2005) focus on leveraging the social graph i.e., the information about the social ties between users in social networks in order to improve recommendations. This is done by enhancing the similarity computation, or by inferring trust between users, assuming that users would prefer recommendations from people they trust. These studies suggest various methods to incorporate trust into the recommendation process. For example, Golbeck (2006) selected only raters with the highest trust values for the similarity computations, while (Massa & Avesani, 2007) base their similarity measures on the level of trust between users. Other systems use friendship rather than trust, for example (Guy et al , 2009, Liu et al., 2010). (Guy et al., 2009) coin a new term: "familiarity" that is extracted from users' relations and activities on social networks and show that recommendation of software items based on a familiarity network is more effective than the recommendation based on the collaborative similarity between users.

In the current study we do not aim at showing that utilizing the social relations between users as a source of information improves recommendations, (as it is a well-known finding already), but rather we make use of the content that users publish and vote on in their Facebook pages to derive their preferences. We use the social relations only as a tool for crawling the network and reach additional users.

Recent studies (Yifan et al., 2008, Koren et al., 2009, Koren 2011) derived users preference data from her behavior on e-commerce sites, such as purchase information, to replace explicit rating for recommender systems. Results show that it is feasible to utilize such data for recommendations. The e-commerce behavior data is similar to

Facebook voting information. For example, (Ben-Shimon et al., 2007) utilized data from a defined social network of a media application to derive the users' interest, and (Das et al., 2007) uses users clicks on a link as indication for their interest in the news item that it represents. Some other studies (e.g. Kumar et al., 2008, Hayes et al., 2007) exploit users' tags to infer their level of interest in items they tag. .

Amer-Yahia et al (2009) suggested a theoretical logical architecture that is aimed at managing and discovering content from social sites and network considering the content as well as the social relations. However, the framework was not yet implemented or tested for enhancing recommender systems.

Another approach that integrates content from social networks into the recommender process is presented by (Wang et al., 2010). In their SocConect project, they recommend users and activities to social networks members, blending information from the various social networks that the user is registered to. However, their system includes an explicit rating mechanism where the user is asked to provide explicit feedback for the recommended activities by annotating them (and by providing like/dislike indications). This approach is quite different from ours which utilizes data that is available anyway and does not involve asking the user to explicitly provide data for the recommendation process. In SoNARS (Carmagnola et al., 2009), the authors infer data regarding the level of a user's interest in an item from a social network by analyzing the behavior of other users with the candidate item. The score of relevance of an item for a user is measured by her friend's activity on the item and the degree of friendship between the user and that friend. The degree of friendship is based on psychological analysis of users' social behavior on a social network. The study that is presented in this paper suggests a much simpler method to derive the user's interest in an item by inferring the user's own opinion from her published content without the need of computing friendships.

A recent work by (Ma et al., 2011) extracts preferences data as well as friendship from the user's social network. While this work is similar to ours in the idea of extracting preference data from the network, they extract explicit rating information (by using epinions (www.epinions.com) and Duban (www.duban.com) that contain explicit ratings on items), while we infer the user's interests from the user's published content as indication of their preferences rather than ratings. In addition, they use

preference information only in the domain of recommendation rather than preferences of user in other domains, as we do, to allow cross-domain recommendations.

Regarding the extraction of information specifically from Facebook to improve recommender systems, we were able to identify very few studies that actually dealt with Facebook data for that aspect. One is the study by (Bourke et al., 2011) that conducted a user study with 82 participants using a Facebook application to recommend items to users while leveraging the social graph focusing on different methods to form the neighborhood. Unlike our study that used Facebook data to infer user preferences from her published content, Bourke et al. follow the line of study that utilizes the social graph for the formation of the user neighborhood for accurate computation of the similarity between users. Another system that used Facebook as a test bed for experiments is SoNARS (Carmagnola et al., 2009) that was mentioned above. SoNARS recommends Facebook groups to Facebook users based on the preferences of the users' friends that are inferred according to their activities on the groups. The recommendation algorithm considers the strength of the relations between users that is defined by utilizing psychological theories, and is based on the nature of interaction between users. Our method for recommendation with Facebook data is much simpler and easy to implement as it infers user interest in an item only from the user's own published content. In addition, we also apply cross domain data that can be useful when data in the required domain is missing.

Yet another recommender system that used Facebook data is a Facebook application for news items recommendation developed by (Agrawal at al., 2009). This system is not personalized for users but for communities and makes use of users' explicit rating as well as click through data on recommended items. The system does not extract information from the users' profile as is done in our study but rather uses the feedback gathered within the application.

2.2 Cross Domain recommendations

Cross-domain recommendations are gaining popularity due to the growth of available cross-domain data in social networks and in e-commerce sites that sell different types of items of many domains and wish to prevent the need for users to provide separate feedback for every domain they purchase from. In addition, cross-domain

recommender systems may adhere to results in marketing research (e.g., Lariviere et al., 2004) that highlight the effectiveness of promoting products from different domains to a user if they fit her buying patterns. For example, some users tend to buy trendy, expensive, cheap, or popular products across all domains. In addition, cross-domain recommendations may increase the diversity of products that a user can receive and might increase revenue.

Some recent studies propose several techniques for cross-domain recommenders when certain conditions apply. For example, Winoto and Tang (2008) examine the relatedness of domains while others aim at constructing a unified user model across domains (Berkovsky et al., 2007, 2008; Tsunoda & Hoshino, 2008) while tackling the user model interoperability challenge (Carmagnola et al., 2011). Interoperability refers to the ability of identifying and using user's data across applications despite differences in the data formats, or languages used, or any other difference.

Another line of research applies machine learning methods, specifically, transfer learning, to enable cross-domain applications that do not depend on the existence of overlapping users between domains (Li et al, 2009; Cao et al., 2010). Initial results reported by these studies reveal the potential for improvement and the challenge of using cross-domain data for recommendation. However, most of the studies mentioned above conducted experiments with no real cross-domain data. For example, Berkovsky et al. (2008) used the EachMovie dataset that was partitioned to artificial domains by genres due to lack of cross-domain data. (Li et al. 2009; Cao et al., 2010); (Shi et al., 2011) performed limited experiments with data from two domains. However, they assumed that there were no overlapping users between domains (or at least their identities across domains were not known). (Li et al. 2009; Cao et al., 2010) used sophisticated transfer learning rather than simple aggregations of ratings while others (e.g., Shi et al., 2011) tried to identify users across different systems, for example by assuming that users assign tags to items and look for mutual tags between domains.

For social network data this assumption is not necessary since a user reveals her information in many domains, thus, overlapping between domains is known information that should be utilized and the user model interoperability challenge is

actually avoided. Our study is the first to test the cross-domain effect on real cross-domain data with overlapping users and to examine the effect of different parameters of the dataset on recommendation results.

3. Research questions

Our study empirically investigated the following research questions:

1. The feasibility and effectiveness of utilizing available preference data from Facebook to replace or complement rating data. To investigate this issue we compared results obtained by collaborative filtering (CF) of rating data with those obtained using data about the users preferences derived from Facebook accounts.
2. The effect of using cross-domain preferences crawled from a social network to recommend items in a single target domain. The users' preferences data from various domains (TV shows, musicians, movies, etc.) that is available on the network is used to analyze effects of cross-domain data on recommendation results for specific domain pairs. We also investigate the influence of several dataset characteristics on accuracy of results.

The main motivations for the above goals are: (a) to take advantage of data that is freely available on social networks as a means of tackling the problems of users who do not cooperate in providing ratings and as a means to enrich the user profile; (b) to address the sparsity and cold-start problems that are common in CF-based recommenders by utilizing data from the networks in the domain of recommendation or from other domains to enrich sparse data or to build an initial user profile for cold start scenarios.

Facebook and other social networks are gaining popularity (at April 2012 Facebook numbered almost 850,000,000 users[2]). Thus, the data aggregated in social networks like Facebook can be leveraged for recommender systems or any other personalization

[2] www.**facebook**.com/press/info.php?statistics

systems that requires knowledge about users. In social networks like Facebook, users reveal considerable information about their preferences, feelings, activities, etc. This information can be very valuable in determining the actual needs and preferences of users.

4. Recommendation strategies

Table 1 summarizes the recommendation strategies examined in this paper. We included strategies that make use of different types of Facebook data and considered various methods of utilizing the data. We compared the results with two baseline state of the art collaborative filtering algorithms, the first is an improved version of the basic k-NN method and the other is an SVD based method. Following is a detailed description of the recommendation strategies. First we detail the baseline methods, then, the methods that use Facebook mentions about movies followed by methods that use cross domain mentions . [3]

Baseline – CF-NN

This technique is used as one of our baseline methods for comparison. The common CF user-based recommender method that uses the *k* most similar users to an active user (k-Nearest Neighbors or k-NN). Similarity weight *Wau*, between the active user *a* and any other user *u*, is based on Pearson correlation coefficient (presented on equation 1) between the vectors of their mutual rated items (C^+) (Adomavicius, and Tuzhilin, 2005).

(1)

$$w_{au} = \frac{\sum_{\in C^+}(r_{u,i} - \bar{r_u})(r_{a,i} - \bar{r_a})}{\sqrt{\sum_{\in C^+}(r_{u,i} - \bar{r_u})^2 (r_{a,i} - \bar{r_a})^2}}$$

where $r_{u,i}$ denotes the rating of user *u* to item *i* and $\bar{r}_{u.}$ the mean ratings of user *u*.

[3] Exact details about the nature of collected data is presented on section 5.1 where the experiments and the data collection procedures are described.

The prediction of a rating of an item for a user, as defined by the equation 2, is based on the ratings of her neighbors (R^+) to the items, where the degree of similarity between users ($w_{a,u}$) defines their effect on the prediction based for example on the Pearson correlation coefficient between their ratings. We used a maximal number of 50 neighbors and considered only neighbors with similarity weight >0.5.

(2)

$$p_{a,i} = \bar{r}_{a.} + \frac{\sum_{u \in R+} w_{a,u}\left(r_{u,i} - \bar{r}_{u.}\right)}{\sum_{u \in R+} w_{a,u}}$$

In order to improve the predictive performance, we have modified the simple k-NN to include

A. Significance weighting (Herlocker et al., 1999) that helps to reduce the influence of nearest neighbors whose similarity weights are computed based only a few available ratings to common items. In particular we replace the computed weighting with:

(3) $$w'_{a,u} = \frac{\min\{\gamma, \left|I_a \cap I_u\right|\}}{\gamma} w_{a,u}$$

where I_u represents the subset of items that have been rated by user *u*. Following (Herlocker et al., 1999) we are using $\gamma = 25$

B. Shrinkage to the mean (Bohnert and Zukerman, 2009) that is known to improve the accuracy of the estimated ratings. The predicted rating is revised as:

(4) $$\hat{p}_{a,i} = \tilde{r}_{.i}^{a} + \omega(p_{a,i} - \tilde{r}_{.i}^{a})$$

where $\omega \in [0,1]$ is a parameter whose value should be determined using a cross-validation approach. In our case we used the value $\omega = 0.8$. $\tilde{r}_{.i}^{a}$ indicates the baseline rating prediction for user *a* and item *i* by averaging of the deviations from the users' means:

$$\tilde{r}_{.i}^{a} = \bar{r}_{a.} + \frac{\sum_{u \in U_i} (r_{u,i} - \bar{r}_{u.})}{|U_i|} \quad (5)$$

where the notation U_i indicates the subset of users that have rated an item i.

Baseline- SVD

Singular value decomposition (SVD) CF methods transform users and items to a joint latent factor space. SVD has recently become the method of choice when the main task is to predict user's ratings. Therefore we use it as another baseline method for comparison. In this paper we adopt the SVD implementation proposed by (Koren and Bell, 2011), which solves the regularized least squares error problem using a stochastic gradient descent procedure (see Section 5.3.1 in (Koren and Bell, 2011) for a detailed description of the method). In particular the predicted rating is provided by:

$$p_{a,i} = \bar{r}_{..} + b_i + b_a + q_i^T p_a \quad (6)$$

where $\bar{r}_{..}$ indicates the overall average rating. The parameters b_a and b_i indicate the observed deviations from the average of user a and item i, respectively. The vectors q_i , $p_a \in \mathbb{R}^f$ respectively measure the extent to which the user or item relate to each one of the f latent factors.

Preliminary experiments showed that setting the number of latent factors to five (f=5) provides the best results. It should be noted that because our dataset is relatively small (95 users and 150 items) using additional latent factors did not improve results, due to over fitting the training data. Hence, by adding additional latent factors, the mean squared error on the training set continues to drop while the mean squared error on the test set increases. Our dataset is relatively small. Therefore using a complicated model (with many latent factors) can very easily overfit the training data, i.e. having good predictive performance on the training set but very poor performance on the test set. This correlation between small training set and overfitting has been analyzed in the past (see for example Karystinos and Pados, 2000).

Methods that use Facebook mentions related to the domain of recommendation :

Facebook data on the recommended domain - k-NN

Similarity between users is computed using related data to the domain of recommendation (movie) from collected Facebook preferences. We used Jaccard similarity (Equation 7) to compute the similarity instead of Pearson correlation since Facebook ratings are in unary form, and no ratings are available. Thus the similarity between two users, *a* and *u*, is defined as the mutual preferences normalized by the number of their aggregated preferences (Candillier et al., 2008). This method attempts to use only Facebook data and does not involve any numeric rating provided by the user.

$$w_{a,u} = \frac{|I_a \cap I_u|}{|I_a \cup I_u|} \quad (7)$$

where I_u represents the subset of items that have been found in the Facebook profile of user *u*. The system returns all movies that are published by the users' friends with similarity above 0.5 (set empirically).

Facebook Popularity

This method uses only Facebook data. Popularity recommendations are regarded as offering a possible remedy for new user situations in CF (Al Mamunur et al., 2002), i.e., situations where no data about a user is available. In rating-related systems, popular items are usually determined as the top-rated items. The Facebook popularity method simply lists the top-mentioned user preference items in the Facebook profiles. In this paper, the recommended list consists of the top 20% popular items.

Methods that utilize Cross domain data

Facebook Cross-domain data

This method utilizes also only Facebook data. Here, for similarity we use Facebook preferences from all domains (movies, TV shows, and music items). Then, we use the Facebook preferences related to recommended domain (movie) to generate rating predictions for movies based on preferences of users that are similar to the active user in all domains. We combine the information from all domains and generate a new user/item voting matrix that includes items from all domains. We then compute the

similarity using Jaccard similarity (equation 7). We return movie related items that are mentioned by users that were found similar to the active user (similarity >0.5). Since movie related preferences are too sparse (only 5% of the users bother to reveal their movie related preferences), cross-domain preferences using related Facebook data might improve results by adding more data to the similarity process.

Facebook Cross-Domain data for similarity and rating for prediction

This method enriches the ratings with Facebook data. The similarity between users is computed using cross-domain Facebook preferences (movies, TV shows, and music items) using Jaccrd similarity on the combined user/item matrix from all domains. The prediction of the preference of items is computed based on the user's explicit ratings. This method examines a possible combination between ratings and Facebook data when ratings are too sparse to determine accurate similarities between users in one domain. Preference data from other domains is added to form a combined data rating matrix to compute the similarity between users, so that the list of items in the rating matrix is larger containing items from all domains.

On Table 1 we list for each strategy its main objective. The strategies are categorized by the motivation of their application. Two strategies are Baseline for comparison (Collaborative filtering and SVD), three others aim at replacing the ratings (as a remedy for the cold start and sparsity challenges, and for better usability of systems), while the last method aim at enriching the ratings with additional complementary data from Facebook.

Most recommendation strategies that we include consist of two phases: the similarity computation (i.e., identifying the active users closest friends), and the prediction. Each of these phases can use different data as input that is listed on the Table as well as the method that is used to operate the phase.

Table 1: Summary of recommendation strategies

Motivation	**Algorithm**	**Input for similarity**	**Similarity Computation method**	**Prediction input**	**Prediction method**
Baseline	Baseline-CF-NN	Ratings	equation 3	Ratings	equation 4
	Baseline-SVD	Ratings	N/R	Ratings	equation 6
Rating Replacement	Facebook-data on the recommended domain (movies)	Facebook mentions of movies	Jaccard similarity	Facebook mentions on movies	Return all items suggested by users with similarity>0.5
	Facebook data-Popularity	Facebook mentions on movies	N/R	Facebook mentions	Return top 20% of popular items
	Facebook data Cross –domain	Facebook mentions in all relevant domains	Jaccard similarity	Facebook mentions	Return all items suggested by users with similarity>0.5
Rating enrichment using Facebook data	Facebook data cross –domain and ratings	Facebook mentions	Pearson correlation	Ratings	Equation 4

5. Experiments and results

5.1 Data

Data collections consisted of two separate procedures. The participants of the first procedure were 95 students at the Faculty of Engineering at Ben-Gurion University in Israel in their third and fourth year of study.

During the first collection procedure, we collected explicit user ratings on 170 popular movies via a specially developed Web form. The movies were selected based on lists of popular movie that were published in popular movies sites. The participants were asked to rank 150 popular movies (on a scale 1-5). The value "1" meant "do not like the movie"; and the value "5" meant "love the movie". The participant could also indicate "0" meaning that she is "not familiar with the movie", which indicate that no

rating is provided. On average, each user rated 80 movies, and each movie was rated by 55 users (sparsity was 65%).

During the second data collection procedure, we developed a crawler that crawled the participants' accounts and was able to obtain data from user profiles about the user preferences of music items, TV series and movies, and about the users' first and second degree friends. The participants on this data collection phase were the same users that participated in the rating procedure. These users agreed to install the application in their Facebook accounts. Seventy five out of the ninety-five students participants in the rating procedure owned Facebook accounts (as of March 2010 when the collection took place). These participants actually formed the seeds for the crawl. Thus, we were able to access their first and second degree friends. It should be noted that on the time the described experiment was conducted users tended to have less friends than it is expected nowadays. In particular, in our experiment the users had on average 84.3 direct friends (standard deviation of 51, min value of 9, max value of 245).

Our crawler implemented breadth-first crawling in which we first access the profiles of the direct friend and only then the second degree friends (FOAF -friends of a friend). We limited our crawling to the first 160 crawled profiles for each user in order to ensure that crawling time will not exceed 5 minutes as promised to the participants when they agree to join the experiment, and to comply with Facebook's automated data collection terms which limit the number of pages that can be downloaded per minute. Due to these constraints the average number of indirect friends was 56.8 and a standard deviation of 55.2.

For each user we collected information from up to their 160 friends (first and seconddegree friends). Altogether we obtained data from a social sub-network which includes 7700 profiles (nodes) and 10,000 friendship ties (edges), where 550 ties are between the original 75 participants. Based on the above network characteristics we can derive the graph density which is defined as the ratio of the number of actual edges and the number of maximum possible edges in the graph. For our dataset the graph density is 3.4 x 10-4 which is in the same order of other social networks that were analyzed in the literature (see for example (Yang et al. 2012)).We also have data

about 750 different movies, 4600 music items and 650 TV series. We found that only 5% of the users revealed information about their movie preferences, while 30% of the users revealed information about TV series and music items they liked. Thus, the sparsity for the movies domain was 0.9948.

Since Facebook data is generated by users with no control on the quality or the format and structure of the content, an accurate extraction of positive mentioning of preferred items requires a sophisticated application that will apply methods such as sentiment analysis to identify positive mentioning, sophisticated stemming and spell checking in order to identify variations and errors for the titles of movies. As our goal in this research was not to develop methods for the analysis and extraction of Facebook data, we avoided this problem by collecting all information to a Database that included a lot of "garbage" and duplicate information. For example, a user could indicate a movie in her fan page and write something about the same movie in her status page). We applied simple heuristics, namely simple n-gram, to identify duplicates and spelling errors. We then used a controlled manual process on the results and "cleaned" duplicates and noise leaving only the relevant information, i.e. titles of movies, music items, and names of TV series. For simplicity we considered similarly one or multiple mentions of an item, as a positive indication without applying any weighting scheme.

To extract information about user preferences for the above domains (movies, music items and TV series), we looked at the following information from the users' accounts: the information in her profile; pages that the user declared as being a fan of (Facebook had the fan option back in 2010 but does not have this feature today); and links that the user published that contained relevant keywords. For example we looked for known titles of movies or music items that were in our set of movies. We looked for keywords like "movie" and semantic related keywords such as film, actor, director etc. In addition, we examined statuses and updates that a user published and that contained relevant keywords.

The set of semantically related keywords were set and adjusted manually, as well as the results of the analysis that were controlled manually. The outcome of this process was for each user a set of preferred items in the domains of interest (i.e., movies, music, TV) that was aggregated to a special database that we prepared.

To test our results we applied repeated random cross-validation. Splitting the data randomly to 80% training and 20% test sets for each set of user ratings, we ran 10 iterations of each algorithm, finally averaging the results for analysis.

5.2 Metrics

We applied the following accuracy measures to evaluate results, precision, mean average precision, R-precision and mean absolute error (MAE). We also measured recall (which is actually catalog coverage) to evaluate the ability of the system to find all relevant items. (Shany & Gunawardana 2011).

Precision in our context is defined as the number of relevant items that were recommended to a user from among the recommended items (sometime referred in this context as hit set). For the rating data, we considered as relevant any item that was rated or given a rate of 4 or above (Herlocker 2004, Dahlen et al. 1998). Precision could be computed for all methods that we examined.

Mean Average Precision (MAP) (Herlocker et al., 2004) measures the accuracy of returned results and their ordering (ranking) compared to the users stated preferences, such that relevant items returned higher on the list result in higher MAP :

$$\frac{1}{N}\sum_{i=1}^{N}\frac{i}{p_i}$$

where:

N – is the number of relevant recommendations;

$\frac{i}{p_i}$ – is the precision value at a given cut-off rank pi;

i – is the number of relevant recommendations of rank pi or less.

We used mean average precision in addition to general precision since the latter is biased towards few returned items, while average precision considers also the quality of the ordering of the returned lists. Thus, for example a system that returns 5 relevant items out of 10 would have the same general precision if the relevant items are the first 5 or the last 5 returned where average precision would prefer a result with the 5 relevant items on top of the list.

R-Precision measures the precision at the nth level, where n is the number of active user's preferred items in the test set. Hence, a system that returns n relevant items is ideal since it returns all the users' preferred items. Thus R-Precision measures the difference of the system to the ideal.

MAE is the average error between the ranks, as given by the user, to the rank predicted by the system (the lower, the better) (Herlocker et al., 2004). This metric could not be used for unary data (Facebook preferences) as it measures the error of the rate predicted by the system compared to the specific value of the rate as given by the system.

MAE was measured for methods that involved ratings in their prediction and for methods where Facebook data was utilized solely for inferring the similarity weight between users. Precision can be measured for all methods including those that utilized Facebook data for similarity and prediction. While the methods measured with MAE looked at the benefit of integrating data collected form social network with explicit data, the methods measured with precision evaluate the potential of utilizing Facebook data instead of explicit ratings.

Recall measures the proportion of relevant returned items and the available relevant items. Recall is similar to coverage that measures the ability of the system to return any relevant recommendations (i.e., higher recall means that the system found relevant more items to return).

In order to compare the performance of the methods, we followed the robust non-parametric procedure that Demsar (2006) proposed. First we applied the adjusted Friedman test on the null-hypothesis, that all methods provide the same results. Once the null hypothesis was rejected, we used the post-hoc Nemenyi test in order to compare the methods with each other. We were especially interested to see if there was a significant difference between the Baseline k-NN method and each of the other tested methods.

5.3 Results

Table 2 presents the averaged MAE and precision results where an asterisk (*) indicates a significant result according to the abovementioned hypothesis test. The Baseline–CF-NN was trained with all the data collected. This data display a sparsity level of 65% and it should be noted that this sparsity level is not common and not realistic in real systems. In real recommender systems, the sparsity level tends to be greater than 95% (see, for example, the Netflix data (www.Netflixprize.com). On Table 2 results refer to the rating dataset with 65% sparsity. However, we also show (on Figure 1) results for all the algorithms that utilize rating data with different levels of simulated sparsity (ranging from 0.6 to 0.99). In particular we show the results as compared at 95% which is a minimal common sparsity in commercial systems, and for sparsity 99.84% which is the sparsity observed for the Facebook dataset that we used.

MAE- We compared the three methods for which MAE could be computed. We used the adjusted Friedman test to reject the null-hypothesis, that all methods present the same MAE performance with $F(4,36)= 27.62$, $p<0.001$. Using the post-hoc Nemenyi with $p<0.05$, we can conclude (as observed in Table 2) that none of the methods performs significantly worse than the baseline methods.

Precision - The null-hypothesis, that all six methods for which precision was measured, provide the same result and that the observed differences are merely random, was rejected using the adjusted Friedman test with $F(8,72)= 67.91$,$p<0.001$. Then, using the post-hoc Nemenyi test with $p<0.05$ to compare the methods (especially the baseline method with the others) it can be observed that the only method that is comparable to the Baseline-CF-NN are the SVD-Baseline, is the "cross-domain using ratings". The methods that are based solely on Facebook data perform significantly lower, namely: Facebook single domain data, cross-domain preferences, and the method that used popularity on Facebook. However, the Baseline-k-NN was merely significantly superior to Facebook cross-domain data; if the significance level is set to 0.009, the methods are comparable. Results show the same trend for the other accuracy measures namely average precision and R-Precision.

Recall - The null-hypothesis, that all six methods for which recall was measured, provide the same result and that the observed differences are merely random, was rejected using the adjusted Friedman test with $F(8,72)= 54.63$,$p<0.001$. Then, using the post-hoc Nemenyi test with $p<0.05$ to compare the methods it can be observed that the only methods that are comparable to the Baseline-CF-NN are the "SVD-Baseline", "Facebook Popularity".

Table 2: Averaged results (over ten iterations). An asterisk indicates that the result is significantly different from the baseline CF-NN with $p<0.05$

Method	MAE	Precision	Recall	Mean Average Precision	R-Precision
Baseline-CF-NN	0.841	0.813	0.491	0.525	0.469
Baseline-SVD	0.876	0.801	0.49	0.49	0.43
Facebook-data on recommended domain (movies)	n/a	* 0.714	*0.42	*0.4	*0.33
Facebook Popularity	n/a	*0.72	0.48	*0.27	*0.28
Facebook data cross-domain	n/a	*0.77	*0.4	*0.38	*0.22
Facebook data cross-domain and ratings (for prediction)	0.842	0.794	*0.435	*0.312	*0.346

As initially observed from the above results, using rating data for predicting the users preferences of movies to users seem to be more accurate than prediction based on movies derived from Facebook data. However, we assumed that the superiority of explicit rating based predictions might be caused by the difference of the sparsity level between the rating and Facebook datasets that were compared. For the ratings dataset, sparsity reached a level of 65% while for the Facebook data only 5% of the users expressed movie preferences, and sparsity for these users and the number of movies that were mentioned reached 99.48%. The sparsity level of Facebook cannot be improved or manipulated since it demonstrates actual user behavior on Facebook. However, the sparsity of the ratings dataset is not realistic for commercial movie recommender systems where sparsity is usually higher than 95% (e.g., Netflix). Therefore, we conducted an additional comparison adjusting the level of sparsity of the rating data to a comparable level (95%) to the Facebook data and ran 10 iterations

on which we randomly selected data to eliminate for sparsity simulation. We present the performance of rating based techniques on a range of sparsity levels (0.6-0.99) and demonstrate the effect of sparsity on results. Although missing ratings are not random (Marlin et al., 2007), we believe that the results provide a good approximation of the effect of sparsity on performance. Moreover, in this case the original rating data does not reflect a common user rating scenario since the users were given a list of movies to rate from which they rated all the movies they knew, and not the movies they selected to rate. Thus the argument over the ratings model and missing ratings does not hold for this data collection procedure.

Figure 1 presents precision for different levels of sparsity for the ratings dataset for all rating-based methods, namely CF-NN, SVD, and FB Cross-domain ratings (as explained on Table 1). For Facebook based method, we show the results at the point of the actual sparsity to compare for comparison with the other method at that point. We could not manipulate of course the sparsity levels for the Facebook based data as the data is already very sparse- 99.48%. It can be observed that at the 99.48% point of sparsity (the point that is comparable to the level of Facebook data sparsity), precision for Facebook reaches 0.714 where precision for the explicit rating Baseline is 0.689. For that level of sparsity, cross-domain recommendations, using only Facebook data reaches a precision of 0.77 and outperforms CF-NN baseline. The simple popularity method when compared at the 99.48% point of sparsity, the Facebook movie domain is lower than the Facebook cross domain but are superior to the baseline algorithms and all other rating based methods are lower than the Facebook data based methods. Thus, we can conclude that sparsity has a clear effect on performance, and when techniques based on ratings data and facebook data are tested on the same sparsity level, Facebook data based techniques are at least comparable and even perform better than explicit rating-based data for sparse data.

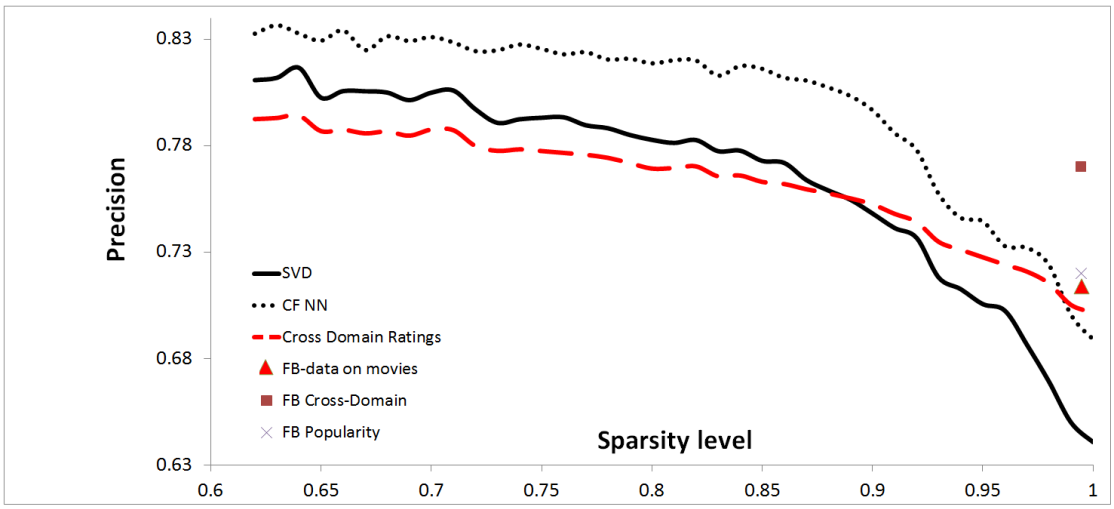

Figure 1 – Sparsity level effect on Precision for the rating dataset

The concavity of the graphs allows us to conclude the interesting finding that starting from sparsity level of 99% "Cross Domain Ratings" outperforms CF NN, probably because CF NN cannot provide any accurate prediction with very little data and it seems better not to operate it on such situations.

5.3. Analysis and Discussion

One encouraging conclusion is the possibility of achieving at least competitive results by replacing the annoying process of acquiring ratings from users with data that is implicitly available from social networks. The sparsity of the Facebook data in our dataset for the movie domain is higher than typical sparsity on commercial systems that is usually around 95%-98%. Thus we show on Figure 1 that the precision for the Baseline k- NN and SVD at the sparsity level of 95% is 0.74 and 0.7 respectively. Both precision values are lower than the precision that can be obtained using Facebook preferences (0.77). Nevertheless when we compared the recall performance, Baseline k-NN at the sparsity level of 95% obtains recall of only 0.426 while SVD obtains a much higher recall of 0.65. Facebook data yields recall of 0.46.

We have to qualify the results by the fact that CF-NN was trained over a rating dataset that involves a relatively small number of total users. One would expect that in a larger rating dataset the results can be improved because there are a greater variety of potential similar users.

Cross-domain data seems to achieve higher results than single domain. Since such cross-domain data is freely available in social networks (e.g., Facebook, MySpace) where people state their preferences in various domains, we conducted more experiments to further investigate the conditions under which it is beneficial to use cross-domain data (see section 6.2).

It should be noted that predictions based on popularity of movies among Facebook members reached a similar level of precision as the Baseline k-NN method for 95% sparsity (which is a realistic sparsity level in real world systems). Thus, it seems rather feasible to use Facebook popularity data for new user cold-start situations. Moreover, if a user does not have yet any history in some domain, our results show that using popularity information from the social network would provide pretty satisfying results, with a precision of above 70%, and might be even higher if a larger set of data is available to compute popularity.

6. Cross-Domain additional experiments

The initial experiment using cross-domain data that was described above, demonstrated the potential of using cross-domain Facebook data to improve accuracy of recommendations. We therefore explored this issue in more details, trying to define the parameters that affect the performance of cross-domain data and the best methods for utilizing them.

We wanted to examine several possible aggregation methods from multiple domains that could improve the accuracy of recommendations in the target domain (i.e., the domain for which the recommendations are required). In addition, we examined the effect of the size of the dataset (number of users), the extent of overlapping users, and the relatedness between domains. Since we could not collect explicit ratings from users for all the examined domains, we compared the cross-domain Facebook results to Facebook results obtained from one domain (i.e., preferences that users explicitly stated). We were then able to define the conditions under which recommendations based on cross-domain data may outperform recommendations based on a single domain. To enable unbiased analysis of the effect of the domain features we apply

cross-domain aggregation only on domain pairs rather than on many domains, so that effects may be isolated.

To base our results on more than one dataset, we performed one more crawl of Facebook accounts and collected more Facebook reference data, starting with another seed of 37 users, mainly graduate and some undergraduate engineering students at Ben-Gurion University from whom we received permission for the crawl. It should be noted that because the second set was crawled a few months after the first set, we needed to update our crawler application in order to adjust to the changes that Facebook introduced in their web-site. We reached 4125 users from which we extracted information from their profiles and fan pages. This time we focused on three domains that included the largest number of items, namely: movies, TV series and musicians, and looked at all available domain pairs (i.e., six variations). To enable meaningful training when data was partitioned for training and testing, we included in the dataset only users who had at least 10 mentioned preferences in each domain that we examined. Thus, from the new crawled data (referred to hereafter as dataset1) we were able to extract 400 users, and from the 7000 users of the first crawl (referred to hereafter as dataset2), we were able to extract only 1000 users that adhered to the above requirements. In Tables 3 and 4 we present information about the data in the two datasets, including the number of users in each domain; number of votes; sparsity; and overlapping users between each pair of domains.

Table 3: Dataset1

Domain	# of votes	# users	# items	sparsity	#overlap users with film	#overlap users with TV shows
Musicians	5463	266	2006	0.98976	25	63
TV shows	1662	100	360	0.95383	23	all
Film	641	34	330	0.94287	All	23

Table 4 : Dataset2

Domain	# of votes	# users	# items	sparsity	#overlap users with Film	#overlap users with TV shows
Musicians	20046	810	3887	0.99363	58	114
TV shows	2570	164	457	0.96571	40	all
Film	1446	65	456	0.95121	All	40

Below we define the cross domain aggregation methods. We denote D_T as a target domain for which predictions are generated while D_S is a source domain for available information.

1) Combine aggregation method

This method consists of merging two domains into one combined preference voting matrix and computes the recommendation for the target domain using the combined matrix so that the list of items on the rating matrix consists of items from all domains. This method might work for sparse target domains where the data from the source domain adds information. (We used this method in our preliminary experiments that were described earlier on section 4).

2) Weighted k-NN

The set of K_S nearest neighbors and their similarities from domain D_S is computed, as well as set of nearest neighbors from domain D_T (K_T). The two sets of nearest neighbors are combined into the set of the most similar neighbors K. The intuition is that collaborative recommendations should be based on the opinions of the users who

are most similar to the current user even if the similarity derives from another domain. However, we give different weights to neighbors from the source and target domains (giving higher weight to data that is based on neighbors in the target domain). We examined the effect of the weighting parameter, and looked at the following weights (10-90, 50-50, 30-70). Similarity between an active user c and a user x is thus defined as:

(8) $$sim\ (c,x)=w(D_S)*simD_S(c,x)+w(D_T)\ *simD_T(c,x)$$

where

$w(D_S)$, $w(D_T)$ – are relative weights given to source and target domains, $w(D_S)$ and $w(D_T)$ complement to 1. This method is relevant for overlapping users, i.e., users who have votes in both domains and might benefit for more data from either domains.

3) k-NN source (denoted as k-NN-s) aggregation method.

In this method, the set of nearest neighbors K_S (from the set of overlapping users) and their similarities are computed in the source domain D_S and are used in the target domain D_T in order to generate a prediction for an active user who appears in both domains. This method might be beneficial to users who are new to domain D_T but are known in domain D_S.

As a baseline method, we used local k-NN that was performed on the target domain to compare the combined and the weighted k-NN methods. The k-NN source is suited for a new-user situation in the target domain. Consequently, local k-NN is not a comparable method since it would not work without data about the target domain. We thus compared the weighted k-NN to recommendations based on the popularity of movies in the target domain. This is a common remedy (Adomavicius and Tuzhilin, 2005) for the new user problem in collaborative filtering.

We ran experimental simulations to compare the performance of the various cross-domain aggregations. For each pair of domains one has been referred to as the source domain and the second as the target domain. By default, the entire source domain was included in the training set. As for the target domain, we employed 10 folds cross-validation on the user level. Namely, in each fold, the entire target data of the 90% of

the users were included in the training set. As per the remaining examined users (10%) we included only portion of their target data in the training set depending on the examined scenario:

- The existing user scenario: In this scenario we assumed that the examined users already had some activities in the target domain, thus, for each user we randomly choose 50% of their declared interests in the target domain to be included in the training set. The remaining 50% of their declared interests in the target domain formed the test set.
- The new user scenario: In this scenario we assumed that the examined users had no activity in the target domain. Therefore, the users' entire target data was not included in the training set, but it constituted the test set.

We used the same evaluation metrics that we used on the first experiments described above, namely precision, mean average precision (MAP), R-precision and recall (since precision and MAP follow the same trend, we present only results for MAP as it considers also the quality of ordering which is relevant for recommendation results); MAE is not relevant since Facebook data is *unary*, and no ratings are available.

6.1 Cross domain additional results

We first present results for the combine and k-NN weights aggregation methods vs. the local k-NN baseline method (section 6.1.1) and then present results for the k-NN source aggregation method vs. popularity (section 6.1.2).

6.1.1 Comparison of combine and local k-NN

As initial findings, we present results averaged over all domain pairs and the two datasets for each of the metrics (Figure 2). Then, we present a detailed analysis that looks into the particulars of the results. Figure 2 presents results of the combine and k-NN weights methods vs. the baseline k-NN applied on the target domain (denoted as local). It refers to the scenario of existing users who has already been active in the target domain. As can be observed, results are not consistent across the metrics. The combine method slightly outperforms the baseline for R-precision and recall and is

slightly inferior for average precision (not significantly different, as explained below) The k-NN weights method was inferior to local for the average precision; performs equally when measured by R-precision; and is superior in terms of the recall measure. It is also apparent that the weight of the weighted k-NN does not seem to have any effect on the results as all weights present similar results.

The overall results are not consistent across measures. We first look at the significance of the results and then analyze other aspects of the results with deeper granularity to understand these differences. We believe that for our data-averaged results, as presented above, there is little value in understanding the actual behavior of a system that might be sensitive to specific domains and features of the dataset. It is interesting to note that reported values are relatively low. The average precision and recall extend at their highest point to about 0.4, and R-precision is even lower (the highest point is ~0.23), i.e., the systems are very far from their ideal point. This can be explained by the high sparsity of the training data.

To analyze the significanse of the above results we performed a non-parametric Friedman test to detect if there are differences between the tested methods for the different datasets and domain pairs. A Friedman test was applied separately for each metric. As input for the Friedman tests, we took the results from five methods: Local-kNN as the baseline, combined, and weighted k-NN with 3 weight variations (10-90, 50-50, 30-70) from 12 blocks of data (6 domain pairs on 2 datasets. The null hypothesis (that there is no difference between methods) is rejected for mean average precision and recall

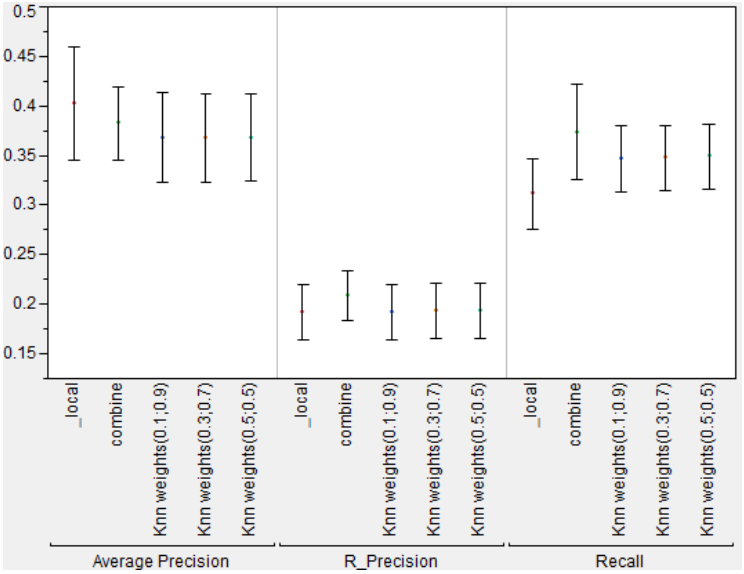

Figure 2 : Combine vs. k-NN Weights

Once the null hypothesis is rejected we apply the Bonferroni post-hoc procedure (Demsar 2006) for pairwise comparisons between the methods to identify the methods that are significantly different from each other (specifically between the local and the cross-domain methods) for the mean average precision and the recall metrics for which significant differences were identified. Results are shown on Table 5. For the mean average precision metric, the results indicate that the combine and local do not exhibit significantly different performance, while the k-NN weights underperform local. For the recall metric, the local method is significantly inferior to both k-NN and the combine method.

Table 5: Results of post hoc procedure. * presents a significant result

Method1	Method2	Mean Difference avg. precision (method1, method2)	P-value Mean avg. precision	Mean Difference Recall (method1, method2)	P-value Recall
Local	Knn weights(0.1;0.9)	2.46	0.0139*	-3.1	0.0019*
Local	Knn weights(0.3;0.7)	2.85	0.0044*	-3.22	0.0013*
Local	Knn weights(0.5;0.5)	2.85	0.0044*	-4.14	<0.0001*
Local	Combine	0.91	0.3628	-5.03	<0.0001*

The above results are averaged over the two datasets and for all domain pairs. It might happen that the results of specific domain differ from the rest due to specific features

of the domain. Thus, we wanted to examine results on a deeper granularity for specific domains and datasets. We applied a paired t-test to examine the particular factors. The results for the difference between the combine and local are presented on Table 6 along with results of a significance test for the hypothesis that combine is better than local for every domain pair and for each dataset. Each column relates to a different measure (mean average precision, r-precision, and recall), and each row relates to a different domain pair. For example, the first row in the table refers to the source domain "film" in dataset1 and target domain "music" in dataset1. For each row we provide the t-test results for each metric separately.

Table 6: Paired t-tests results for Combine > Local

Domain pairs D_S , D_T	**Mean avg. precision p-value H1:** $\mu_{Combine} > \mu_{Local}$	**R-Precision p-value H1:** $\mu_{Combine} > \mu_{Local}$	**Recall p-value H1:** $\mu_{Combine} > \mu_{Local}$
Dataset1			
Film,mus	0.3117	<0.0001*	<0.0001*
Film,tel	0.5843	<0.0001*	<0.0001*
Mus,film	<0.0001*	<0.0001*	0.0002*
Mus,tel	0.1154	0.0049*	<0.0001*
Tel,film	0.1469	0.0015*	<0.0001*
Tel,mus	0.9998	0.5632	<0.0001*
Dataset2			
Film,mus	1.0000	0.0076*	<0.0001*
Film,tel	0.9999	0.0542	<0.0001*
Mus,film	0.9949	0.0396*	<0.0001*
Mus,tel	1.0000	0.9291	<0.0001*
Tel,film	0.6543	0.3260	0.4985
Tel,mus	1.0000	0.9397	<0.0001*

In accordance with the general results on Table 5, the combine method does not improve the results significantly for the average precision aspect (only for one domain in one dataset). But it does improve recall (more relavant items are found). Table 6 also highlights difference in the performance results for R-Precision (that were not

significantly different as a whole) for most of the domain pairs in dataset1 and only for 2 domain pairs in dataset2.

To perform a similar paired t-test for k-NN weights method vs. the local method for specific domain pairs and datasets, we wanted first to ensure that there is no difference between the weights as can be estimated from Figure 2 and Table 5. This would make it possible to compare only one best version of the k-NN weights method. The ANOVA measures that we applied to the three versions of k-NN weights yielded no significant differences, i.e., the different weights of the k-NN weights method did not significantly affect the results. Thus, for further tests we examined only one variation of the k-NN weights method – similar weights (0.5 for source and 0.5 for target). Paired t-tests for each domain pair and dataset adhered to the overall results (on Table 5). For mean average precision, there was no significant difference for most domain pairs on the two datasets. For R-Precision there was a difference between the datasets: the k-NN method was inferior for most domain pairs on dataset1 and not significantly different on dataset2. Recall for k-NN source was superior in both datasets for most of the domain pairs.

One clear conclusion is that cross-domain methods are able to obtain more relevant items (recall) but fail to provide significantly more accurate results than the local method across all domain pairs. It is also notable that the combine method is the better cross-domain method compared to k-NN weights across all measures.

The superiority of the cross-domain method in obtaining more relevant items (recall) can be also perceived in Figure 3 that shows the average number of actual user votes for each domain and dataset and the average number of relevant items obtained by each of the methods for each domain and the two datasets (please note that since for each pair of domains we consider only the overlapping users between domains, the number of votes for a single domain is different when paired with different domains, e.g., film when paired with music has a different number of votes from when it is paied with television). Obviously, the cross-domain methods consistently obtain more relevant items than the local, but due to high sparsity they do not find more than half of the relevant items in most cases.

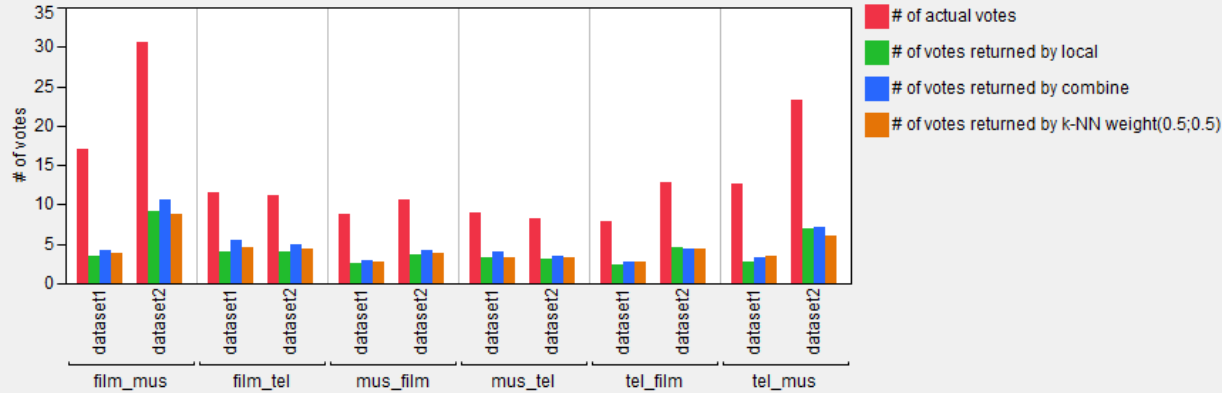

Figure 3: Number of actual votes and of relevant items obtained by each method

Due to the fact that different domains have different sparsity levels, the reported results are provided in different scales. In an attempt to normalize the scale, we also performed relative analysis. We used mean percentage error (MPE) to quantify the relative difference between the examined aggregation method and the baseline for each metric, normalized by the baseline method that reveals the trends of results (equation 9). Positive MPE means relative superiority of the examined method, emphasizing cases having low precision.

(9)

$$MPE = \frac{1}{n}\sum_{i=1}^{n}\frac{CrossMethod_i - BaselineMethod_i}{BaselineMethod_i}$$

To demonstrate the effect of the relativeness of the equation, assume two scenarios: On the first, the cross method obtained 0.1 precision and the baseline method 0.05. The MPE result is 100% since the precision of the cross method is doubled relative to the baseline methodas the result is 100% better. On the second scenario, the precision of the cross method was 0.25 and the baseline was 0.2, the MPE is 25%. Note that in both cases, absolute performance improvement was 5% which does not reflect the difference between the scenarios. Relative improvement comparison is common in machine learning and a version of the MPE was used in (Opitz & Maclin, 1999). MPE is important for observing trends when results are low, as is our case.

MPE for the Combine method:

MPE was positive through all domain pairs and for both datasets for the three evaluation metrics as presented in Figure 4. Here, when a relative comparison is applied for each data set and domain pair, the results are consistently positive, but present high variance of results between datasets and domains. However, it can be concluded that the combine method consistently improves recommendation results, but the degree of improvement varies between datasets, domain pairs and measures. The highest improvement was observed for average precision (in 7 out of 12 cases), while the lowest for r-precision (in 9 out of 12 cases). The differences between R-precision and average precision adhere to the findings reported by (Aslam et al., 2005) that for precision values under 0.5, R-Precision is underestimated while average precision is overestimated.

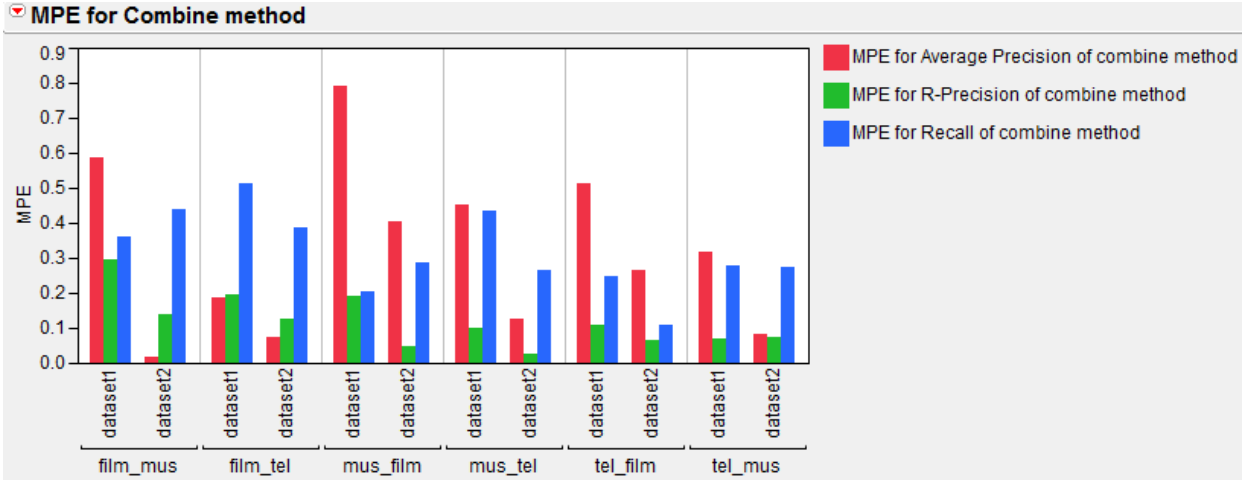

Figure 4: MPE for Combined.

MPE for Weighted k-NN:

A comparison between the weight levels of k-NN indicated that there is no difference between the levels, thus we compared only one level (50,50) to the baseline. As observed from Figure 5, for the two precision metrics and for both datasets, the MPE for weighted k-NN with the baseline did not show a steady pattern of improvement. Mean average precison results are improved in most cases (10 out of 12) , but R-precision results are not improved in 7 out of 12 cases. Recall consistently improves for all cases. The improvement for precision in Dataset1 is more visible

than in Dataset2. This might be explained by the fact that Dataset1 is sparser for all domains as observed from Tables 3 and 4 that presented the datasets' features.

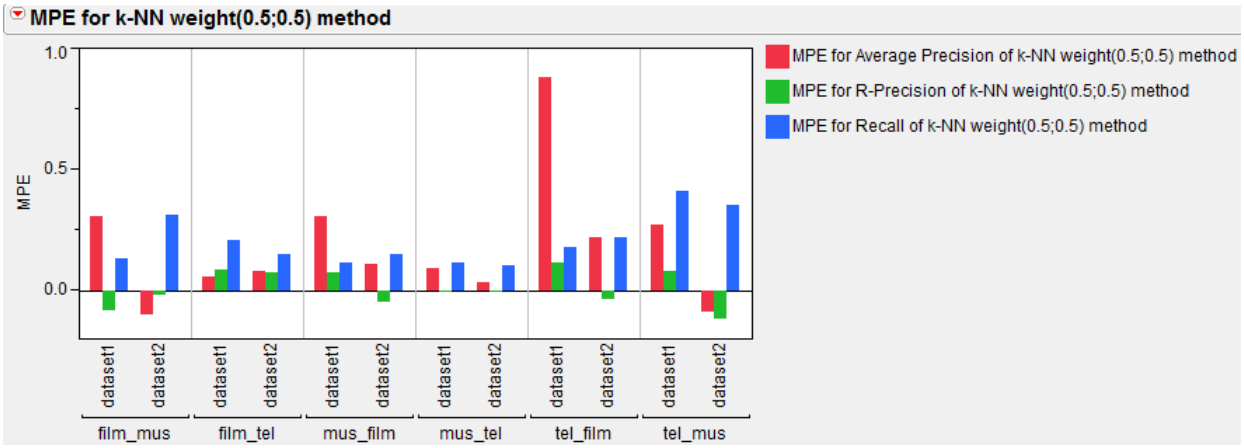

Figure 5: MPE for k-NN Weight (0.5,0.5)

As a summary of the results so far, we can conclude that it seems beneficial to apply cross-domain methods to improve accuracy. We also observe clearly higher recall (that is higher even for absolute measures). Recall pertains to the extent of returning more results even if they do not fit exactly the users' expectations but are more serendipitous. Higher recall is extremley important when the set of relevant items is relatively small since it enables the system to return a sufficient number of relevant items to the user. Since the improvements are sensitive to domains, we later analyze domain features that might explain the differences between domains.

6.1.2 k-NN source and popular results

We now describe results for the k-NN source aggregation method that learns from the source domain the users' neighborhoods and applies what it is has learnt to the target domain. This method aims at alleviating the problem of cold start user in the target domain. We compare it to popularity as a baseline for the new user remedy. The overall results, averaged over domains and datasets, are presented in Figure 6. They show that k-NN is higher for mean average precision. The result for recall would not be considered in the analysis, since the popularity method considers any item that is higher than 0 as a popular item that can be recommended (but ranked accordingly). Thus popularity by definition would not miss any relevant item. For the same reason we do not consider R-precision in the analysis since it is biased towards recall.

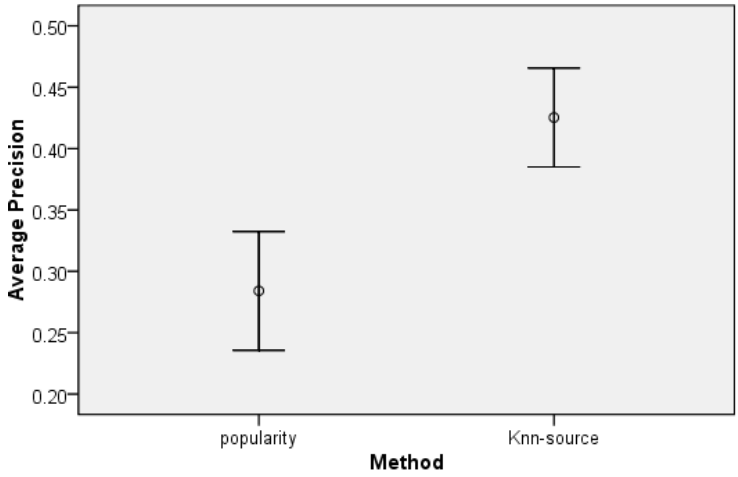

Figure 6: k-NN source vs. popular results

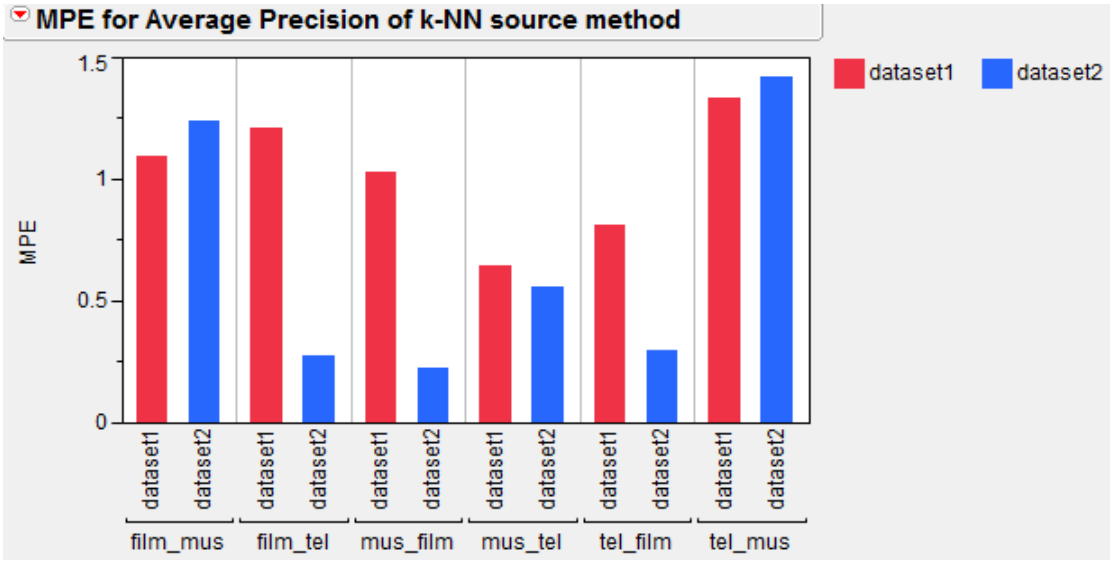

Figure 7: MPEs of k-NN-s method for avg. precision for k-NN source

Figure 7 presents the MPE results obtained for mean average precision for the k-NN source method. As can be observed, k-NN-source consistently outperforms popularity.

We also applied paired t-tests to evaluate the differences for popularity and k-NN source for all domains in the two datasets. In Table 7 we present the results of paired

t-tests for mean average precision. Results are consistent with MPE: the mean average precision for the k-NN-source method significantly outperforms the popularity method for all cases.

Table 7: Paired t-test for mean average precision, k-NN-s vs. popularity

Domain pairs D_S - D_T	**t-Ratio**	**p-value** **H1:** $\mu_{Knn\text{-}s} > \mu_{popularity}$
Dataset1		
film_mus	5.6210	<0.0001*
film_tel	3.4089	0.0013*
mus_film	3.4047	0.0012*
mus_tel	6.1534	<0.0001*
tel_film	4.5235	0.0001*
tel_mus	6.3195	<0.0001*
Dataset2		
film_mus	8.6396	<0.0001*
film_tel	4.7375	<0.0001*
mus_film	3.1907	0.0012*
mus_tel	6.4541	<0.0001*
tel_film	3.3513	0.0009*
tel_mus	11.1939	<0.0001*

We conclude that as a means of solving the problem of lack of information for new users in the target domain, cross-domain is achieving significantly more accurate results than the popularity baseline method.

6.2 Domains features

We can conclude from the results so far that as a general trend, the combine method is relatively superior to the local baseline method for relative measures and for recall, while for the k-NN-weight the superiority is not clear. As for k-NN source vs. popularity, it is also superior for the new user situation. However, these superior features can be observed only as a trend and are not consistent for all datasets and domain pairs. To understand the causes of these differences, we examined the effect of various data parameters on performance. We looked at the effect of the number of

users in each target domain, and the amount of overlapping users between domain pairs. In addition, we examined the correlations between each pair of domains (denoted as $cor(D_T,D_S)$). The correlation is based on the average correlations over all pairs of items which is considering the rating vectors from these domains as rated by overlapping users (Berkovsky et al., 2007). This enables the examination of the effect of relatedness of domains on results. We then analyzed the correlation between the values of these parameters and the mean improvement of the cross-domain method for the average precision and the r-precision metrics. Some correlation results are presented in Tables 8, 9 and 10. An additional feature that we looked at is the effect of the number of items that users mention in the target domain, (which is an aspect of sparsity), on the results, as well as the effect of the ratio between the number of mentions in the source and target domain. Our intuition is that when the number of items in the source domain is relatively higher than those in the target domain, the improvement is more effective.

Table 8: Correlations between domain relatedness and improvements for different measures

Variable1-Data set feature	**Variable2-Improvement Cross-domain method-baseline, Measure**	**Correlation**	**Significance Probability**
$cor(D_T,D_S)$	combine-local, MAP	0.4227	0.1710
$cor(D_T,D_S)$	k-NN weights-local, MAP	0.4623	0.1302
$cor(D_T,D_S)$	combine-local, R-Precision	0.6625	0.0189*
$cor(D_T,D_S)$	k-NN weights-local, R-Precision	0.5874	0.0446*

From Table 8 we observe a significant positive correlation between pairs of domains for the improvement of the cross-domain method on the R-precision metric for both methods. As the domains are more correlated, cross-domain is more effective. This result is very important since it indicates when it is beneficial to apply the cross-domain method.

Table 9: Correlations between number of users and improvements for different measures

Variable1-Data set feature	Variable2-Improvement cross domain method-baseline, Measure	Correlation	Significance Probability
#users D_T	combine-local, MAP	-0.6931	0.0125*
#users D_T	kNN weights-local, MAO	-0.8753	0.0002*
#users D_T	combine-local, R-Precision	-0.4061	0.1902
#users D_T	kNN weights-local, R-Precision	-0.7647	0.0038*

The number of users in the target domain has a significant effect, i.e., a negative correlation is noted between the average difference in the performance and the number of users in the target domain for all methods and for most metrics (except the case of R-Precision of combine-local). The more users in the target domain, the lower is the improvement.

Table 10: Correlations between number of overlapping users and improvements for different measures

Variable1-Data set feature	Variable2-Improvement cross domain method-baseline, Measure	Correlation	Significance Probability
#overlap users	combine-local, MAP	-0.7042	0.0106*
#overlap users	kNN weights-local, MAP	-0.6222	0.0307*
#overlap users	combine-local, R-Precision	-0.8292	0.0009*
#overlap users	kNN weights-local, R-Precision	-0.7349	0.0065*

Table 10 reveals that there is a significant negative correlation (p=0.05) between the mean improvement and the amount of overlapping users. This observation is true for all aggregation methods and for all metrics. We believe that this result is rooted in the fact that as the number of overlapping users grow, the number of users in general and

their ratings in the target domain also grows, and performance for the local improves due to more data.

Table 11 presents an analysis of the effect of the number of items that users mention on the target domain on results. We performed the analysis on all methods with all measures, but we present as an example the results for k-NN weights measured by precision and recall, (absolute and MPE), since all the other results follow the same trend.

Since the maximal number of items per user did not exceeds 60 (for most users), we divided the users based on the amount of items they mention on the target domain to three groups: Small (<20 items), Moderate (20-40 items) and Large (>40 items – mostly up to 60 items). The values presented on the table are averaged over both datasets and over all target domains. A one way Multivariate Analysis of Variance (MANOVA) was used to examine the effect of the independent variable (amount of items user mentions in the target domain) on the two objective criterion variables: precision and recall (by using their absolute values). Results of the MANOVA indicated significant differences between the various items levels with $F(4,193) = 17.4$, $p < 0.01$. As expected, we can see that the precision increases when the user has more items in the target domain. But at the same time, the MPE decreases with the number of items. This indicates that the relative improvement of cross domain is more prominent in cases where the number of items in the target domain is small.

Table 11: Precision and Recall for amount of item mentions in the target domain

Amount of items user mentions in the target domain	Precision		Recall	
	Absolute	MPE	Absolute	MPE
Small (<20)	0.414	0.290	0.3679	0.2475
Moderate (20-40)	0.504	0.075	0.3120	0.1792
Large (>40)	0.524	0.032	0.2246	-0.1760

Table 12 presents the precision and recall according to the ratio of the number of items mentioned in the source domain to the number of items mentioned in target domain: Small (<1), Moderate (1-2) and Large (>2). The values are mean values over both datasets and over all target domains. We can see that the precision increases with the ratio. A one way Multivariate Analysis of Variance (MANOVA) was used to examine the effect of the independent variable (Ratio) on the two objective criterion variables: precision and recall. Results of the MANOVA indicated significant differences between the various ratio levels with $F(4,193)=9.2$, $p< 0.01$. This indicates that the relative improvement of cross domain is more prominent in cases where the number of items in the source domain is larger than the target domain.

Table 12: Precision and Recall as a function of the ratio of mentions in source and target domains

Ratio	Precision		Recall	
	Absolute	MPE	Absolute	MPE
Small (<1)	0.393	-0.0251	0.2803	0.1703
Moderate (1-2)	0.447	0.1304	0.3488	0.2885
Large (>2)	0.509	0.1282	0.3859	0.2714

We can conclude that the performance of the cross-domain is sensitive to various dataset features, namely relatedness of domains, number of users, number of items mentioned and overlapping users. Thus, before using Facebook cross-domain data, the dataset should be analyzed and the recommender should be calibrated accordingly (as in many other recommender systems).

7. Conclusions, Limitations and Future Work

In this study we explored the possibility of utilizing derived user preferences from public social networks specifically from Facebook for collaborative recommender systems. We examined data related to the domain of recommendation and to other

related domains. This information can then be integrated as input to the collaborative recommender system process during the similarity and the prediction phases. This is done for substituting explicit ratings in the new user case when no ratings are available, or in order to avoid user interference. The data from Facebook can also enrich sparse explicit ratings data to improve the performance of the recommender.

Our results demonstrate the potential of the exploitation of available data in Facebook for the improvement of personalized services that require data about users in order to provide effective results. Our findings reveal that recommendations based solely on Facebook data are at least as good as those based on explicit ratings and are even more accurate in certain situations, namely for high sparsity, or when no data is available for generating recommendations, such as cold start situation, and when cross domain data is available. In any case cross domain methods are superior in recall, i.e., they find more relevant items without sacrificing accuracy. This result is practically important as it enables operation of collaborative filtering based recommenders when rating data is sparse or non-existent.

We examined several methods to include cross-domain data from Facebook to improve recommendations and were able to determine some dataset related features that affect performance, such as number of overlapping users, number of items that the user mention on the target and source domains and others. We consider these results as preliminary indications of the benefits of cross domain data integration to the recommender system process and believe that further research is required involving more datasets to identify the boundaries of the cross domain contribution, i.e., when is it beneficial to apply cross domain, how related should the domains be, how sparse etc. Cross-domain should be further studied for advanced methods of data integration, such as aggregating data from multiple domains (rather than pair-wise cross domain, as was done on the current study), considering the degree of relatedness between domains for their effect on the process, or the application of cross domain data from several systems when no overlap between users is known or available. In a continuing project we currently develop machine learning methods to learn the relatedness between domains and patterns of users' behavior on each domain.

Our study had some limitations that are rooted in the fact that it is a user study setting that is known to have biasing limitations (Shany and Gunawardana, 2011). However, an on-line evaluation was not feasible for the purposes of our study given the motivation of comparing many algorithms for the same data and in the same setting. In addition this setting allowed us to obtain explicit ratings from the same users that then installed an application that allows us to crawl their Facebook account and extraction of data from it. We were therefore able to compare explicit ratings with Facebook implicit extracted data.

Another limitation of the study is the size of the datasets and the homogeneity of the population involved. (all seed participants that provided ratings are engineering students in the same university). We believe that larger datasets from diverse users would have yield more significant results of the same trend. However, due to the need to obtain ratings and to install a crawling application at the users' account, a larger dataset was not feasible. . However, we do believe that our results are at least a good approximation to real world settings with larger datasets. Now that we have obtained our controlled results it is justified to perform experiments with real systems that can utilize our findings.

In our study we used a crawler application that our participants installed so that we could crawl their Facebook accounts and extract their and their friends' data, and access all their information under their agreement. Some of the information within Facebook is not publically available and cannot be obtained without the users' cooperation and confirmation due to users' privacy limitations. However, in this study we obtained the users' permission to install an application within their accounts. The data was collected automatically and was stored anonymously in our databases. Of course, crawling and extraction of data should be done with careful preservation of users' privacy regulations. However, privacy issues are beyond the scope of this study.

In addition, we did not deal in our study with efficient automatic extraction of Facebook data but applied simple heuristic procedures with relaxed assumptions (such as the positive nature of content published by users), followed by manual control. However, for a real application, this issue should be handled and studied and sophisticated procedures such as sentiment analysis should be applied. Last but not

least, the list of the types of data that is analyzed and derived from the social network can be extended to include for example context data (e.g., location and time) or information about groups and of course friendship and other social data. We believe that our paper with its encouraging results will serve as a motivation for further research in the field.

Acknowledgements:

We would like to thank Hagit Liven and Shiran Azarya for their contribution to the experiments conducted in this project, to their help in the developments of some algorithms and running some experiments. Without their devotion the above results could not have been achieved.

8. References

Adomavicius,G., Tuzhilin, A . 2005. Toward the next generation of recommender systems. A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, pp. 734-749.

Agrawal, M., Karimzadehgan, M., and Zhai, D. (2009). An online news recommender system for social networks. SIGIR-SSM.

Al Mamunur, R., Istvan, A., Cosley,D., Lam, S.K., McNee, S.M., Konstan,J.A., and Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. *In* Proceedings of the 7th international conference on Intelligent user interfaces (IUI '02). ACM, New York, NY, USA, pp. 127-134.

Amatriain, X., Pujol, J.,Oliver, N. (2009). I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems. In User Modeling, Adaptation, and Personalization; Lecture Notes in Computer Science, Springer Berlin. Pp. 247-258

Amer-Yahia, S. Lakshmanan, L. and Yu. C. SocialScope. (2009). Enabling information discovery on social content sites. In CIDR, 2009

Arazy, O. Kumar, N. and Shapira, B. (2010). A Theory-Driven Design Framework for Social Recommender Systems. Journal of the Association of Information Systems (JAIS), Vol 11(9), pp. 455-490.

Aslam ,J.A. , Yilmaz, E., and Virgiliu P. (2005). A Geometric Interpretation of R-precision and Its Correlation with Average Precision. SIGIR (2005). pp. 573-574.

Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisles, A., Shani, G., Naamani, L. (2007). Recommender System from Personal Social Networks. 5th Atlantic Web Intelligence Conference, pp. 47-55.

Berkovsky, S., Kuflik, T., and Ricci, F. (2007). Distributed collaborative filtering with domain specialization. Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 40.

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

Bohnert, F., and Zukerman, I. (2009). Non-intrusive personalisation of the museum experience, in Proceedings of the 17th International Conference of User Modeling, Adaptation, and Personalization (UMAP 09), pp. 197-209.

Bourke, S., McCarthy, K., Smyth, B. (2011). Power to the people : exploring neighbourhood formations in social recommender systems. RecSys '11 : Proceedings of the fifth ACM conference on Recommender systems

Candillier, L., Meyer, F., and Fessant, F. (2008). Designing specific weighted similarity measures to improve collaborative filtering systems. Advances in Data Mining.Medical Applications, E-Commerce, Marketing, and Theoretical Aspects, pp. 242-255.

Cao, B., Liu, N., and Yang,Q. (2010). Transfer Learning for Collective Link Prediction in Multiple Heterogeneous Domains, the 27th International Conference on Machine Learning, ICML 2010.

Carmagnola, F., Vernero, F., and Grillo, P.(2009). Sonars: A social networks-based algorithm for social recommender systems. In Procedings of the 17th International Conference on User Modeling Adaptation, and Personalization, (UMAP 09), pp. 223-234.

Carmagnola,F., Cena, F., and Gena, C. (2011. User model interoperability: a survey,User Modeling and User-Adaption Interaction (UMUAI). Vol 21(3). pp. 285-331.

Das, A. S., Datar, M., Garg, A., Rajaram, S. (2007). Google News Personalization: Scalable Online Collaborative Filtering. 16th International Conference on World Wide Web, pp. 271-280

Dahlen, B. J., Konstan, J. A., Herlocker, J. L., Good, N., Borchers, A., And Riedl, J. (1998). Jump-starting movielens: User benefits of starting a collaborative filtering system with "dead data". Technical Report 98-017. 1998. University of Minnesota

Demˇsar ,J., (2006). Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research, Vol (7), pp. 1-30.

Golbeck, J., and Hendler, J. (2006). Inferring binary trust relationships in Web-based social networks ACM Transactions on Internet Technology (TOIT) , Vol. 6(4) November 2006, pp. 497-529.

Groh, G., and Ehmig, C. (2007). Recommendations in taste related domains: collaborative filtering vs. social filtering. In Proceedings of the 2007 international ACM conference on Supporting group work (GROUP '07). ACM, New York, NY, USA, 127-136.

Groh, G., Birnkammerer, S., Köllhofer, V. (2012). Social Recommender Systems, in Kacprzyk, J.,Jain, L.C. Recommender Systems for the Social Web. Springer, Berlin Heidelberg.

Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., and Ofek-Koifman, S. (2009). Personalized recommendation of social software items based on social relations. Proceedings of the third ACM conference on Recommender systems (RecSys '09). pp. 53-60.

Hayes, C.; Avesani, P.; and Veeramachaneni, S. (2007). An analysis of the use of tags in a blog recommender system. In Veloso, M. M., ed., IJCAI, pp. 2772–2777.

Huang, C., and Gong, S. (2008). Employing rough set theory to alleviate the sparsity issue in recommender system. International Conference on Machine Learning and Cybernetics, pp.1610-1614.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS). Vol. 22(1), pp. 5-53.

Herlocker, J.L., Konstan, J.A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In: SIGIR ’99. Proceedings. of the 22nd Annual Int. ACM SIGIR, pp. 230–237. ACM, New York, NY, USA.

Karystinos, G.N. and Pados, D.A.(2000).On overfitting, generalization, and randomly expanded training sets, IEEE Transactions on Neural Networks, Vol 11(5). pp. 1050-1057

Koren, Y. and Bell, R.; Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer , Vol. 42(8), pp.30-37.

Koren, Y., and Sill, J. (2011). OrdRec: An ordinal model for predicting personalized item rating distributions, accepted to Recsys 2011.

Koren, Y., and Bell, R. (2011). Advances in Collaborative Filtering, In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul Kantor, editors, Recommender Systems Handbook, Springer, pp. 145-186.

Kumar Vatturi, P., Geyer, W., Dugan, C., Muller, M.,and Beth Brownholtz. (2008). Tag-based filtering for personalized bookmark recommendations. In Proceeding of the 17th ACM conference on Information and knowledge management (CIKM '08)

Lariviere, B. and Van den Poel, D. (2004). Investigating the Role of Product Features in Preventing Customer Churn by Using Survival Analysis and Choice Modeling: the Case of Financial Services Expert Systems. with Applications Vol. (27), pp. 277-285.

Lekakos, G., and Giaglis, G. (2007). A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. User Modeling and User-Adapted Interaction 17(1), pp. 5-40.

Li, B., Yang, Q., and Xue, X. (2009). Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction. In Proceedings of the 21st international Jont Conference on Artifical intelligence (Pasadena, California, USA, July 11 - 17, 2009). H. Kitano, Ed. International Joint Conference On Artificial Intelligence. pp. 2052-2057

Liu, F, and Lee. H. J. (2010). Use of social network information to enhance collaborative filtering performance. Expert Syst. Appl., Vol. 37(7). pp. 4772–4778.

Ma, H., Zhou, D., Liu, C., Lyu, M.R., and King, I. (2011). Recommender systems with social regularization. In Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11). pp. 287-296.

Marlin, B.M., Zemel, R.S.,Roweis, S., and Slaney, M. (2007). Collaborative filtering and the missing at random assumption, Proceedings of the 23rd Conference of Uncertainty in Artificial Intelligence, 47, pp. 50-54

Massa, P. and Avesani, P. (2007). Trust-aware recommender systems, in

Proceedings of the 2007 ACM conference on Recommender systems (RecSys 2007) .pp. 17-24.

Opitz , D., and Maclin , R. (1999). Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence Research Vol. (11). pp. 169-198.

Ricci, F., Rokach, L., Shapira, B., (2011). Introduction to Recommender Systems, in the Recommender Systems Handbook, Ricci, F., Rokach, L., Shapira, B.,Kantor, P.B., Springer US. pp. 1-38.

Said, A., de Luca, E.W., Albayrak, S.(2010). How social relationships affect user similarities. In: Guy, I., Chen, L., Zhou, M.X. (eds.) Proc. of 2010Workshop on Social Recommender Systems (2010)

Shani, G., and Gunawardana, A. (2011). Evaluating recommender systems, in the Recommender Systems Handbook, Ricci, F., Rokach, L., Shapira, B.,Kantor, P.B., Springer US. pp.257-298.

Shi, Y., Larson, M., and Hanjalic, A. (2011). Tags as Bridges between Domains: Improving Recommendation with Tag-induced Cross-Domain Collaborative Filtering. 19th International Conference on User Modeling, Adaptation and Personalization.

Spertus, E., Sahami, M., and Buyukkokten,O. (2005). Evaluating similarity measures: a large-scale study in the orkut social network. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD '05). pp. 678-684.

Tsunoda, T., and Hoshino, M. (2008). Automatic metadata expansion and indirect collaborative filtering for TV program recommendation system. Multimedia Tools and Applications , Vol. 36(1). pp.37-54

Inc. Netflix. Netflix prize. Available at Netflixprize.com

Victor, P. and De Cock, M. and Cornelis,C. (2011). Trust and Recommendations. in Recommender Systems Handbook, Ricc., F., Rokach, L., Shapira, B., & Kantor, P. Ed Springer, USA, pp. 645-672.

Wang, Y., Zhang, Jand Vassileva, J. (2010). Towards Effective Recommendation of Social Data across Social Networking Sites, Artificial Intelligence: Methodology, Systems, and Applications, (AIMSA), Lecture Notes in Computer Science, 2010, Springer Berlin / Heidelberg, pp. 61-70.

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. Computers Vol. 6(3), pp. 209-225.

Yang, C. and Harkreader, R. and Zhang, J. and Shin, S. and Gu, G. (2012). Analyzing Spammers' Social Networks For Fun and Profit – A Case Study of Cyber Criminal Ecosystem on Twitter" accepted to WWW'12

Yifan H., Koren, Y., and Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback Datasets, Data Mining, 2008. ICDM '08, pp.263-272.

Yildirim, H., and Krishnamoorthy,M.(2008). A random walk method for alleviating the sparsity problem in collaborative filtering. In Proceedings of the 2008 ACM conference on Recommender systems (RecSys '08). ACM, New York, NY, USA, pp. 131-138.

Yuan, W.;, Shu, L., Chao, H.C., Guan, D., Lee, Y.-K. and Lee, S. (2010). ITARS: trust-aware recommender system using implicit trust networks, Communications, IET , Vol.4(14), pp.1709-1721.