Exploiting Social Information in Pairwise Preference Recommender System

Crícia Z Felício, Klérisson V R Paixão, Guilherme Alves, Sandra de Amo, Philippe Preux

Exploiting Social Information in Pairwise Preference Recommender System

Crícia Z. Felício[1,2], Klérisson V. R. Paixão[2], Guilherme Alves[2], Sandra de Amo[2], Philippe Preux[3]

[1] Federal Institute of Triângulo Mineiro, Brazil
cricia@iftm.edu.br
[2] Federal University of Uberlândia, Brazil
{klerisson, guilhermealves, deamo}@ufu.br
[3] University of Lille & CRIStAL, France
philippe.preux@inria.fr

Abstract. There has been an explosion of social approaches to leverage recommender systems, mainly to deal with cold-start problems. However, most of the approaches are designed to handle explicit user's ratings. We have envisioned Social PrefRec, a social recommender that applies user preference mining and clustering techniques to incorporate social information on the pairwise preference recommenders. Our approach relies on the hypothesis that user's preference is similar to or influenced by their connected friends. This study reports experiments evaluating the recommendation quality of this method to handle the cold-start problem. Moreover, we investigate the effects of several social metrics on pairwise preference recommendations. We also show the effectiveness of our social preference learning approach in contrast to state-of-the-art social recommenders, expanding our understanding of how contextual social information affects pairwise recommenders.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Clustering-Information filtering; J.4 [**Computer Applications**]: Social and behavioral sciences

Keywords: Pairwise Preferences, Social Network, Social Recommender System

1. INTRODUCTION

Social recommender (SR) could shatter the barriers for users to consume information. Thus, there has been an explosion of social approaches in this context [Tang et al. 2013], mainly to deal with cold-start problems. Typical SR systems assume a social network among users and make recommendations based on the ratings of the users who have direct or indirect social relations with the target user [Jamali and Ester 2010]. However, explicit user's ratings may not capture all the user's interests without loss of information [Balakrishnan and Chopra 2012]. Pairwise preference learning shows clear utility to tackle such problem [de Amo and Ramos 2014]. The "marriage" between pairwise preference recommender and social network can be used in a unique manner to enhance the recommender effectiveness.

In such way, we advance earlier work, PrefRec [de Amo and Oliveira 2014], a model-based hybrid recommender system framework based on pairwise preference mining and preferences aggregation techniques. We propose Social PrefRec, an approach to incorporate social networks information in recommendation task to minimize user cold-start problem. Different factors of social relationships have influence on users. Some of these factors contribute or even harm social recommender systems

C. Z. Felício would like to thank Federal Institute of Triângulo Mineiro for study leave granted. K. V. R. Paixão and G. Alves are sponsored by scholarships from CAPES. We also thank the Brazilian research agencies CAPES, CNPq and FAPEMIG for supporting this work. Ph. Preux' research is partially funded by Contrat de Plan État Région Data, and the French Ministry of Higher Education and Research, and CNRS; he also wishes to acknowledge the continual support of Inria, and the exciting intellectual environment provided by SequeL.

[Yuan et al. 2015]. Understanding the extent to which these factors impact SR systems provides valuable insights for building recommenders. We aim to investigate the role of several social metrics on pairwise preference recommendations. Given that user's preference is similar to or influenced by their connected friends [Tang et al. 2013], we also aim to study how to apply social similarities in a pairwise preference recommender. Social PrefRec is evaluated on two datasets, named Facebook and Flixster, to verify the integrity of our results. Focusing on social pairwise preference recommendation, our study addresses six questions:

Q1: How accurately does social information help on item recommendation?

We will assess the accurateness of Social PrefRec by comparing it to PrefRec. This is the key to determine whether a pairwise preference recommender can benefit from social information.

Q2: How relevant are the recommendations made by a social pairwise preference recommender?

One of the main reasons for the relevance of Social PrefRec is to mitigate the cold-start problem for users through social information. To further assess our model, we compare Social PrefRec to three state-of-art social recommenders.

Q3: Which social metrics are the most important for item recommendation?

The previous questions focus on understanding whether pairwise recommenders could benefit from contextual social information. Here, we want to evaluate the overall performance of each social metric: friendship, mutual friends, similarity, centrality and interaction.

Q4: How effective is Social PrefRec *to mitigate data sparsity problems?*

In social recommender systems there is a common assumption that contextual social information mitigates data sparsity problems. To assess our model in this context, we evaluate the effectiveness of Social PrefRec with regards to PrefRec against five data sparsity levels.

Q5: Does social degree affect Social PrefRec *as much as profile length affects* PrefRec*?*

To achieve high-quality personalization, recommender systems must maximize the information gained about users from item ratings. The more ratings a user's profile has, the merrier will be. We want to check whether increasing the number of friends impacts our approach.

Q6: Are there major differences between recommendations quality of popular and unpopular users?

Here we further investigate social popularity effects on recommender systems. This question complements Q5, offering valuable insights into when and which social metric impacts the predictions.

This article is a follow up to our earlier study of social information on pairwise preference recommendation [Felício et al. 2015], which tackled only *Q1* and *Q2*. Here, we extend this study by revisiting how *Q2* is addressed, and introducing *Q3, Q4, Q5* and *Q6* to give thoughtful discussions about the practical implications of our findings for pairwise recommender systems.

The remainder of this article is structured as follows: Section 2 presents the background knowledge undertaking in this work and review related work. Section 3 describes our proposed framework the Social PrefRec, as well as the applied social metrics and recommender model selection strategies. Section 4 describes our experimental settings and Section 5 presents the results. Finally, Section 6 concludes the article.

2. BACKGROUND AND LITERATURE REVIEW

In this section, we introduce the main concepts underlying this work. Section 2.1 presents the notion of pairwise preference recommender systems. Please refer to de Amo et al. [2013] for more details on

Table I. Movie attributes.

Item	Title	Decade	Director	Star	Genre
i_1	Gangs of New York	2000	Scorsese	Di Caprio	Drama
i_2	Catch me If You Can	2000	Spielberg	Di Caprio	Drama
i_3	The Terminal	2000	Spielberg	Tom Hanks	Drama
i_4	The Departed	2000	Scorsese	Di Caprio	Thriller
i_5	Shutter Island	2010	Scorsese	Di Caprio	Thriller
i_6	Saving Private Ryan	1990	Spielberg	Tom Hanks	Drama
i_7	Artificial Intelligence	2000	Spielberg	Haley J. Osment	Drama
i_8	Bridge of Spies	2010	Spielberg	Tom Hanks	Drama

pairwise preference mining. Following, Section 2.2 describes the related work on pairwise systems and Section 2.3 reviews the literature concerning social recommenders.

2.1 Pairwise Preference Recommender Systems

Let $U = \{u_1, ..., u_m\}$ be a set of users and $I = \{i_1, ..., i_n\}$ be a set of items, $RU(A_1, ..., A_r)$ be a relational scheme related to users, and $RI(A_1, ..., A_t)$ be a relational scheme related to items. The user-item rating matrix in a system with m users and n items is represented by $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ represents the rating given by user u on item i. Table I shows a set of 8 items (movies) and their attributes. A user-item rating matrix with 7 users and movies ratings in the range $[1, 5]$ is illustrated in Table II.

In traditional recommender systems, the recommendation task is based on the predictions of the missing values in the user-item matrix. Pairwise preference recommender systems predicts the preference between a pair of items with missing values in the user-item matrix. Both types of systems use the predictions to extract a ranking of items and recommend the top-k.

In our work, we focus on the PrefRec framework, a hybrid model-based approach to design pairwise preference recommender systems. Essentially PrefRec works in two phases: (A) construction of the recommendation models, and (B) recommendation.

A) Construction of the recommendation models. The main activities of this phase are Preferences Clustering, Consensus Calculus and Preference Mining.

Preferences Clustering: First, PrefRec clusters users according to their preferences. This process applies a distance function and a clustering algorithm $\mathcal{C}$ over the rows of the user-item rating matrix. A preference vector of user u_x is defined as $\theta_{u_x} = R_{u_x}$, where R_{u_x} is a row of matrix R. The output of the clustering algorithm is a set of clusters C, where each cluster C_s has a set of users with the most similar preference vectors.

Consensus Calculus: For each cluster C_s, a consensus operator $\mathcal{A}$ is applied to compute $\hat{\theta}_s$, the consensual preference vector of C_s. $\hat{\theta}_{s,j}$ is the average rating for item j in cluster C_s. Please note that the $\hat{\theta}_{s,j}$ element is computed if and only if more than half of the users in C_s rated the item. Otherwise, this position will be empty.

An example of clustering and consensus calculus can be seen in Table III. The users from Table II were clustered in two groups according to their preference vectors, and a consensual preference vector for each cluster was computed using the group average rating per item.

In comparison with the original PrefRec, one main enhancement done in these two activities, Preferences Clustering and Consensus Calculus, was the replacement of the preference matrix and the consensual matrix by vectors. This new representation not only reduces the algorithm complexity and execution time, but remarkably allows a clustering of a better quality.

Table II. User-item rating matrix.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
Ted	5	2	-	1	-	2	1	-
Zoe	5	2	4	1	5	1	-	3
Fred	4	-	5	-	5	-	1	-
Mary	2	5	3	5	-	-	-	5
Rose	1	-	2	-	2	-	-	4
Paul	-	-	3	4	1	-	-	5
John	2	-	-	5	2	-	-	-

Table III. Clusters of users with consensual preferences.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
Ted	5	2	-	1	-	2	1	-
Zoe	5	2	4	1	5	1	-	3
Fred	4	-	5	-	5	-	1	-
$\hat{\theta}_1$	**4.7**	**2.0**	**4.5**	**1.0**	**5.0**	**1.5**	**1.0**	*
Mary	2	5	3	5	-	-	-	5
Rose	1	-	2	-	2	-	-	4
Paul	-	-	3	4	1	-	-	5
John	2	-	-	5	2	-	-	-
$\hat{\theta}_2$	**1.3**	*	**2.7**	**4.7**	**1.7**	*	*	**4.7**

Preferences Mining: Having the consensual preference vector from each cluster, the system could establish the preference relation between pairs of items. Formally, a *preference relation* is a strict partial order over I, that is, a binary relation $Pref \subseteq I \times I$ transitive and not reflexive. We denote by $i_1 > i_2$ the fact that i_1 is preferred to i_2. According to the previous example, a preference relation over consensual preference vector θ_1 is presented in Table IV.

A preference miner $\mathcal{P}$ builds a recommendation model for each group using item's features. The set of recommendation models is $M = \{M_0 = (\hat{\theta}_1, P_1), \ldots, M_K = (\hat{\theta}_k, P_K)\}$, where K is the number of clusters, $\hat{\theta}_s$ is the consensual preference vector, and P_s is the preference model extracted from $\hat{\theta}_s$, for $1 \leq s \leq K$.

In this scenario, a recommendation model is a contextual preference model. Thus, each model P_s in M is designed as a *Bayesian Preference Network* (BPN) over a relational schema $RI(A_1, ..., A_t)$. A BPN is a pair (G, φ) where G is a directed acyclic graph in which each node is an attribute, and edges represent attribute dependency; φ is a mapping that associates to each node of G a set of conditional probabilities $\mathbb{P}[E_2|E_1]$ of the form of probability's rules: $A_1 = a_1 \wedge \ldots \wedge A_v = a_v \rightarrow B = b_1 > B = b_2$ where $A_1, \ldots, A_v$ and B are item attributes. The left side of the rule (condition event E_1 in conditional probability) is called the context and the right side (condition event E_2 in conditional probability) is the preference on the values of the attribute B. This rule reads: *if the values of the attributes $A_1, \ldots, A_v$ are respectively $a_1, \ldots, a_v$ then for the attribute B the value b_1 is preferred to b_2* . Please note that the preferences on B depend on the values of the context attributes. A contextual preference model is able to compare items: given two items i_1 and i_2, the model can predict which is preferred.

The constructing of a BPN comprehends in: (1) the construction of a network structure represented by the graph G and (2) the computation of a set of parameters φ representing the conditional probabilities of the model. The preference miner used in this work, CPrefMiner [de Amo et al. 2013], uses a genetic algorithm in the first phase to discover dependencies among attributes and then, compute conditional probabilities using the Maximum Likelihood Principle [Nielsen and Jensen 2009].

Example Overview. Considering the relational schema of movie attributes in Table I and the user-item rating matrix in Table II: PrefRec cluster users, extract preference consensual vector, from $\hat{\theta}_1$ (Table III), and builds the pairwise preference relation (Table IV). Then, CPrefMiner can build the BPN depicted in Figure 1. $PNet_1$ represents the contextual preference model that is used to compare the set of pairs of items and make the predictions.

B) Recommendation. In its second phase, PrefRec aims at using a recommendation model M_s to recommend items for a **new user**. It is executed online, in contrast to the first phase which is offline. The recommendation process is executed according to the following steps:

(1) Given a target user u_x and a (small) set of ratings provided by u_x over some items of I, the first task consists in obtaining the consensual preference vector $\hat{\theta}_s$ more similar to u_x's preferences. We compute the similarity between θ_u (the u_x's preference vector) and each consensual preference vector. Let $\hat{\theta}_s$ be the consensual preference vector, related to cluster C_s, the most similar to θ_u.

Table IV. C_1 pairwise preference relation

$(i_1 > i_2)$
$(i_1 > i_3)$
$(i_3 > i_6)$
$(i_5 > i_6)$
$(i_2 > i_6)$
$(i_5 > i_3)$
$(i_2 > i_4)$
$(i_6 > i_7)$

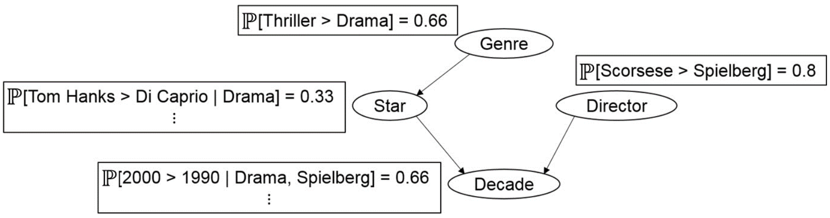

Fig. 1. Bayesian Preference Network $\textbf{PNet}_1$ over C_1 preferences.

(2) Consider the preference model P_s corresponding to $\hat{\theta}_s$. P_s is used to infer the preference between pairs of items in I which have not been rated by the user u_x in the past.

(3) From the set of pairs of items (i_j, i_k) indicating that user u_x prefers item i_j to item i_k, a ranking can be built by applying a ranking algorithm adapted from the algorithm Order By Preferences [Cohen et al. 1999]. Thus, the output is a ranking $(i_1, i_2, \ldots, i_n)$ where an item i_j is preferred or indifferent to an item i_k, for $j < k$ and $j, k \in \{1, ..., n\}$.

Example: To illustrate how a preference model can be used in a recommendation phase, suppose that the preference vector θ_u of a **new user** u_x is most similar to the consensual preference vector of group C_1, $\hat{\theta}_1$. Let us consider the BPN $\textbf{PNet}_1$ built over $\hat{\theta}_1$ and depicted in Figure 1. This BPN allows to infer a preference ordering on items over relational schema $RI(Decade, Director, Star, Genre)$ of data movie setting. For example, according to this ordering, item i_5 = (2010, Scorsese, Di Caprio, Thriller) is preferred than item i_8 = (2010, Spielberg, Tom Hanks, Drama). To conclude that, we execute the following steps:

(1) Let $\Delta : I \times I \rightarrow \{A_i, ..., A_l\}$ be the set of attributes for which two items differ. In this example, $\Delta(i_5, i_8) = \{Director, Star, Genre\}$.

(2) Let $\min(\Delta(i_5, i_8)) \subseteq \Delta(i_5, i_8)$ such that the attributes in $\min(\Delta(i_5, i_8))$ have no ancestors in $\Delta(i_5, i_8)$. According to the $\textbf{PNet}_1$ structure, directed edge linking *Genre* and *Star* implies remove *Star*, therefore, in this example, $\min(\Delta(i_5, i_8)) = \{Director, Genre\}$. To have i_5 preferred rather than i_8 is necessary and sufficient that $i_5[Director] > i_8[Director]$ and $i_5[Genre] > i_8[Genre]$.

(3) Computing the probabilities: $p_1 = \textit{probability that } i_5 > i_8 = \mathbb{P}[Scorsese > Spielberg] * \mathbb{P}[Thriller > Drama] = 0.8 * 0.66 = 0.53$; $p_3 = \textit{probability that } i_5 > i_8 = \mathbb{P}[Spielberg > Scorsese] * \mathbb{P}[Drama > Thriller] = 0.2 * 0.33 = 0.06$; $p_2 = \textit{probability that } i_8 \textit{ and } i_5 \textit{ are incomparable} = 1 - (p_1 + p_3) = 0.41$.

To compare i_5 and i_8 we focus only on p_1 and p_3 and select the highest one. In this example, $p_1 > p_3$ so that we infer that i_5 is preferred to i_8. If $p_1 = p_3$ was true, we would conclude that i_5 and i_8 are incomparable.

2.2 Studies on Pairwise Preference Recommendation

Balakrishnan and Chopra [2012] have proposed an adaptive scheme in which users are explicitly asked for their relative preference between a pair of items. Though it may give an accurate measure of a user's preference, explicitly asking users for their preference may not be feasible for large numbers of users or items, or desirable as a design strategy in certain cases. Park and Chu [2009] proposed a pairwise preference regression model to deal with the user cold-start problem. We corroborate with their idea. They argue that ranking of pairwise users preferences minimize the distance between real rank of items and then could lead to better recommendation for a new user.

In the same direction, Sharma and Yan [2013] propose a probabilistic latent semantic indexing model for pairwise learning, which assumes a set of users' latent preferences between pairs of items.

We build on previous work [de Amo and Oliveira 2014] by adapting a pairwise preference recommender to leverage a graph of information, social network.

2.3 Studies on Social Recommender Systems

This research field especially started because social media content and recommender systems can mutually benefit from one another. Many social-enhanced recommendation algorithms are proposed to improve recommendation quality of traditional approaches [Krohn-Grimberghe et al. 2012] [Alexandridis et al. 2013] [Canamares and Castells 2014]. In terms of using user's social relationships to enhance recommender systems effectiveness, there are some common themes between Social PrefRec and the works of Ma et al. [2008] [2011] [2011]. They developed two approaches called SoRec and SoReg. The former is based on probabilistic matrix factorization to better deal with data sparsity and accuracy problems. The latter also relies on a matrix factorization framework, but incorporates social constraints into its built models.

Furthermore, TrustMF is an adaption of matrix factorization technique to map users in terms of their trust relationship, aiming to reflect reciprocal users' influence on their own opinions [Yang et al. 2013]. SocialMF also explores the concept of trusting among users, but in the sense of propagation into the model [Jamali and Ester 2010]. In comparison, the goal of Social PrefRec is to exploit social networks in pairwise preference fashion. Although we also employ both users' social network information and rating records, we do it in a different way. Instead of embedding social information in recommendation models, we built a loosely coupled approach based on clustering techniques to choose a suitable model for a given user.

3. SOCIAL PREFREC

Social PrefRec proposes a new approach to address the new user problem through social information. It is a PrefRec framework extension, incorporating social information at recommendation phase. There were no modification on how models are built, but at recommendation phase we propose an alternative based on social information to recommend items for new users.

In a simple way, a recommendation for new users using social information could recommend items well rated by his direct friends. Another option is to leverage the connection weight among friends to provide better recommendations. The challenge here is to determine how much influence or similarities exist among user's relationship. Connect weight among users can be computed through similarities on profiles (profession, age bracket, location, etc.), interaction between users (messaging, photos, etc.) and degree of influence.

To support this feature, we extended PrefRec and devise Social PrefRec. Figure 2(a) presents its new structure. To better understand it, let us consider the set of users U and the set of items I aforementioned in Section 2. The weight function $w : U \times I \rightarrow \mathbb{R}$ computes a *user preference degree* for an item and can be represented by a rating $r_{u,i}$ from a user-item rating matrix R. To represent a social network, let $G = (V, E)$ be a social graph, and u_x and u_y vertices of this graph (users of a social network). A set of friends (neighbors) of a vertex u_x is $F(u_x) = \{u_y | u_y \in V \wedge (u_x, u_y) \in E\}$ and a function $l : F \rightarrow \mathbb{R}$ defines *connection weight* between u_x and u_y in $[0, 1]$.

An illustrative example of a social graph in Social PrefRec is shown on Figure 2(b). Nodes represent users, and edges are friendship relations. Edges are labeled with the connection weights computed as explained in Section 3.2. Dashed groups are clusters of users, and each cluster is associated with a recommendation model. Suppose that Paty is a new user; so, there were no historical preferences associated to her. However, the system already clustered Paty's friends according to their preferences. As soon as Paty shows up, the connection weight is computed, and a suitable recommendation model is selected.

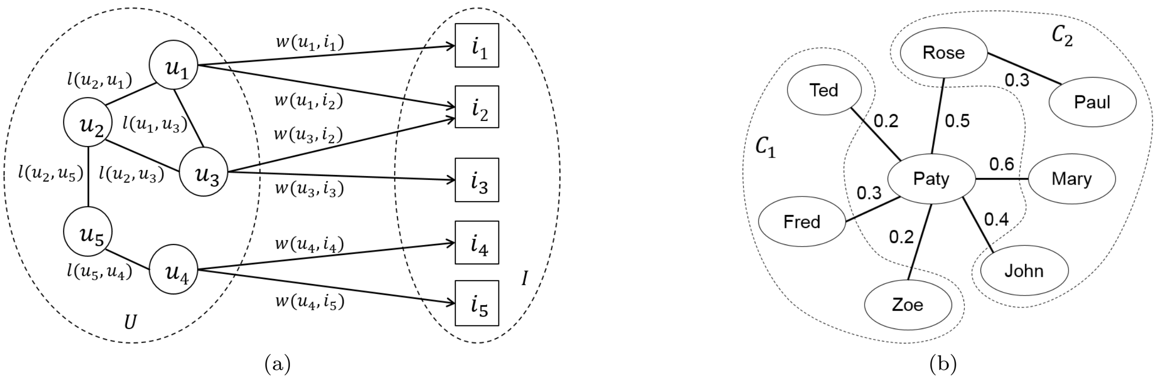

Fig. 2. (a) Social PrefRec structure and (b) Social network example.

3.1 Social PrefRec Framework

The general architecture of Social PrefRec, the interactions among the five modules, as well as their respective input and output are presented in Figure 3. Modules from 1 to 4 are from PrefRec, but in module 5 (Recommendation), Social PrefRec, unlike its predecessor, chooses proper recommendation model using one social metric according to following steps.

(1) Given a target user u_x and a social metric, we will select u_x's friends $F(u_x)$ and the related connection weight, previously computed as described in Section 3.2, between u_x and each $u_y \in F(u_x)$.
(2) Using one of the selection model methods (see Section 3.2), we will select the preference model P_s corresponding to the cluster C_s with more similar friends.
(3) P_s is used to infer the preference between pairs of items in I.
(4) From the set of pairs of items (i_j, i_k) indicating that user u_x prefers item i_j to item i_k, a ranking can be built as mentioned in PrefRec approach.

Note that using this strategy, it is possible to recommend to a given user without taking into account any previous ratings, but relying on the user's relations in the cluster set.

3.2 Computation of connection weights and recommendation model selection

Given the social graph G for a target user u_x and for each $u_y \in F(u_x)$, we compute user's connection weight according the following metrics:

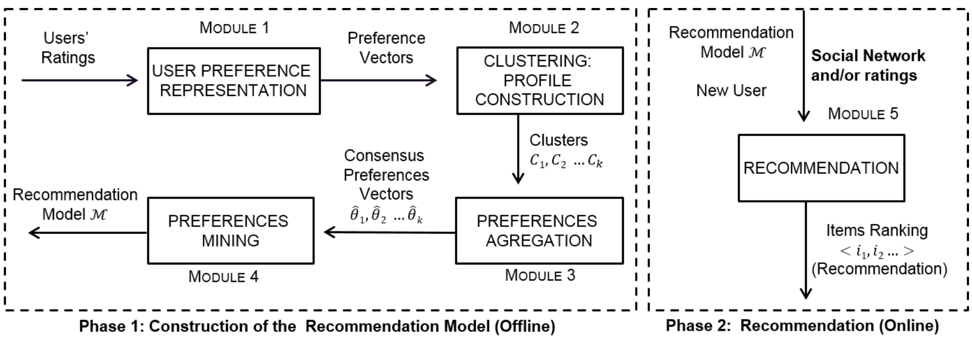

Fig. 3. Social PrefRec Framework.

—*Friendship*: in this metric, connection weight is measured by $l(u_x, u_y) = 1$, where $1(\cdot)$ is the characteristic function (1 if argument is true, 0 otherwise).

—*Interaction level*: computed as $\frac{a(u_x,u_y)}{\widehat{a}(u_x)}$, where $a(u_x, u_y)$ is the number of times that user u_y appears at u_x's time-line, and $\widehat{a}(u_x)$ is the number of all occurrences of users u_y at u_x's time-line.

—*Mutual friends*: Represents the fraction of common friends or Jaccard similarity using $l(u_x, u_y) = \frac{F(u_x) \cap F(u_y)}{F(u_x) \cup F(u_y)}$.

—*Similarity score*: Given by demographic similarity between u_x and u_y according to function $l(u_x, u_y) = sims(u_x, u_y)$. We compute this value by the average of individual similarity in each demographic attribute (Age bracket, Sex, Religion, etc), using the binary function $similarity(u_x, u_y, A_i)$, wich returns 1 if attribute A_i is similar for u_x and u_y, 0 otherwise.

—*Centrality*: Calculated by average of closeness, betweenness and eigenvector centrality measures with $l(u_x, u_y) = centrality(u_y)$.

Social PrefRec allows the definition of any strategy to find a recommendation model. To do so, Module 5 provides a function to select a recommendation model. Let $select : U \rightarrow M$ be a function that selects the proper recommendation model from M for a target user u_x. In this work, Social PrefRec uses two strategies for recommendation model selection based on connection weights: minimum threshold and average connection weight. Each strategy has a different type of implementation for function *select*, as explained in the following definitions:

—*Minimum threshold*: Let $\varepsilon \in [0, 1]$ be a minimum threshold for connection weight. The minimum threshold strategy selects the preference model P_s (associated with model $M_s \in M$) which has more users who have a connection weight with the target user u_x equal or above a minimum threshold according to Eq. (1).

$$select(u_x) = \arg\max_{M_s \in M} |\{u_y \in F(u_x) \wedge l(u_x, u_y) \geq \varepsilon\}| \tag{1}$$

—*Average*: The average strategy selects the preference model P_s with users who have the highest average connection weight with the target user u_x according to Eq.(2).

$$select(u_x) = \arg\max_{M_s \in M} \frac{1}{|F(u_x)|} \sum_{(u_x,u_y) \in F(u_x)} l(u_x, u_y) \tag{2}$$

4. EXPERIMENTAL SETTING

4.1 Datasets

Table V summarizes our datasets. Recall that sparsity is the percent of empty ratings in user-item rating matrix and links are the number of users connections in the dataset. The particularities of each dataset is described next:

Facebook Dataset. We surveyed this dataset through a Facebook web application we developed for this purpose. With volunteers permission, we crawled relationship status, age bracket, gender, born-in, lives-in, religion, study-in, last 25 posts in user's time-line, posts shared and posts' likes, and movies rated before on the Facebook platform. In addition, we asked each volunteer to rate 169 Oscar nominated movies on a 1 to 5 stars scale. We got data from 720 users and 1,454 movies, resulting in 56,903 ratings.

In our experiments, we consider only ratings from the 169 Oscar nominated movies, which represent movies rated by most users. We split Facebook data into two datasets, FB50 and FB100, to represent the set of users who rated at least 50 and 100 movies, respectively. This was done to evaluate the

Table V. Dataset features.

Dataset	Users	Items	Ratings	Sparsity (%)	Rates / User (Average)	Links	Links / User (Average)
*FB*50	361	169	44.925	26.36	124.44	2,926	8.6
*FB*100	230	169	35,459	8.77	154.16	1,330	6.4
Flixster 175*K*	357	625	175,523	26.36	491	706	2.8
Flixster 811*K*	1,323	1,175	811,726	47.78	613.54	6,526	5.34

overall system performance under datasets with different sparsity and social information levels. The movie's attributes are: genres, directors, actors, year, languages and countries. In FB50 and FB100, we compute user similarity metric using the attributes: relationship status, age bracket, gender, born-in, lives-in, religion and study-in. We also compute the interaction level considering the last 25 posts in the user time-line, posts shared and likes.

Flixster Dataset. Jamali and Ester [2010] published this dataset. However, movie information was restricted to its title, then we improved it by adding genres, directors, actors, year, languages and countries information retrieved from IMDB.com public data. We also use two datasets from Flixster with different sparsity level, Flixster 175K and Flixster 811K. Flixster social information includes friend's relationships, mutual friends, friends centrality and users similarities. Similarity between users is computed only through three attributes: gender, age bracket and location. Interaction information is not available on Flixster dataset.

4.2 Comparison Methods

In our experiments we compare Social PrefRec with PrefRec and three social matrix factorization based recommender systems. The idea is to evaluate Social PrefRec recommendations compared to PrefRec. Note that the former chooses the prediction model using only social information whereas the latter needs user's first ratings to choose a model. Further, the comparison with matrix factorization methods is used to evaluate Social PrefRec compared to other social approaches, which handle cold-start users.

Social matrix factorization methods combine social information with rating data. They are distinct from Social PrefRec that uses social information only to choose a consensual prediction model between preference clusters. In addition, our method has its recommendation model based on pairwise preferences. The three social matrix factorization methods do not make use of any clustering technique. We take these systems as comparison methods because they achieve high accuracy levels for cold-start user as reported by the authors. The social matrix factorization particularities are reported next:

SoRec [Ma et al. 2008]: it is based on latent factors of items, users, and social network relationship. The influence of one neighbor on the prediction of a rating increases if he is trusted by a lot of users while it decreases if the target user has many connections.

SocialMF [Jamali and Ester 2010]: applies a trust propagation mechanism. More distant users have less influence (weight) in rating prediction than the trust direct contacts.

TrustMF [Yang et al. 2013]: represents the influence of connections to target user preferences in two ways: truster and trustee. This approach provides recommendations to users that usually show influence on others and those who are typically influenced by others.

4.3 Experimental Protocol

Each experiment was performed on the datasets split into two parts: training set and test set.

PrefRec and Social PrefRec build clusters (K-Means clustering) of similar users using the training set. For each cluster C_s the systems associate a recommendation model M_s. Then, to recommend items for a given user u_x, it is necessary to select the most similar model (cluster) that fits u_x. This process is done during the test phase. However, those approaches take different directions. Since PrefRec is not able to deal with social information, it relies on previous ratings of u_x to select its best recommendation model. In contrast, as Social PrefRec requires social information to accomplish this task. We employ the leave-one-out protocol [Sammut and Webb 2010] to better validate our tests and simulate a realistic cold-start scenario. Thus, for each test iteration, one user is taken for test purpose, and the training set is made of all other users. Each experiment is composed by n iterations, where n is the number of users. Importantly, because PrefRec cannot act in a full cold-start scenario, we give PrefRec a few ratings to bootstrap the system.

PrefRec *protocol.* The PrefRec recommendation model is built offline. For the test phase y ratings of the current test user u_x chosen at random were considered for the choice of the most similar cluster C_s. Then, computing the similarity between the preference vector of u_x, θ_{u_x}, and the consensual preference vector of C_s, $\hat{\theta}_s$ is a matter of computing the *Euclidian distance* between these two vectors weighted by the number of common ratings (z), where $d_E(\theta_{u_x}, \hat{\theta}_i) = \frac{1}{z}\sqrt{\sum_{k=1}^{z}\left(\theta_{u_x,i_k} - \hat{\theta}_{i,i_k}\right)^2}$. Please note that this similarity distance was used for preferences clustering (training) and selection models (test) phases. Finally, for validation purpose, the remaining ratings of the current test user u_x were used.

Social PrefRec *protocol.* Building the recommendation model is done as in PrefRec. However, during the test phase, we do not take any rating into account. Social PrefRec requires solely social information to find the most similar cluster, C_s, according to a given social metric and a model selection strategy.

Matrix Factorization social approaches protocol. For SoRec [Ma et al. 2008], SocialMF [Jamali and Ester 2010] and TrustMF [Yang et al. 2013] the experimental protocol builds a model M_x for each user u_x using friendship preferences information which includes all friend's item ratings. In contrast to previous protocols, the recommendation model, M_x, is not a clustered preference model, but a specific preference model for each user.

Parameter Settings. In our experiments, we use LibRec [Guo et al. 2015] which contains an implementation of SoRec, SocialMF and TrustMF methods with default parameters. We executed Matrix factorization approaches with 10 latent factors and the number of interactions set to 100. We use K-means as the clustering algorithm for PrefRec and Social PrefRec. In addition, we experimentally test several numbers of clusters. Then we set the optimal number of clusters for each dataset: 7 for FB50, 6 for FB100, 4 for Flixster 175K, and 2 for Flixter 811K. The minimum threshold ϵ has optimal values equal to 0.4 for FB50 and FB100, and 0.1 for Flixster 175K and Flixter 811K. However, we executed experiments related with *Q5* and *Q6*, over FB50 and FB100 with $\epsilon = 0.1$ to have more users in the result set to evaluate these two questions.

4.4 Evaluation methods

Regarding our evaluation method, we present results using two metrics: (1) ***nDCG*** is a standard ranking quality metric to evaluate the ability of the recommender to rank the list of top-k items [Shani and Gunawardana 2011]. (2) We also compute the standard F_1 ***score***, based on precision and recall, to evaluate the prediction quality of pairwise preferences [de Amo and Oliveira 2014].

In the ***nDCG*** equation (3), $r_{u,1}$ is the rating (according to the ground truth) of the item at the first ranking position. Accordingly, $r_{u,j}$ is the ground truth rating for the item ranked in position j. M is the number of ranked items. $DCG(u)$ is the discounted cumulative gain of predict ranking for a target user u, $DCG^*(u)$ is the ground truth and N is the number of users in the result set.

$$DCG(u) = r_{u,1} + \sum_{j=2}^{M} \frac{r_{u,j}}{\log_2 j}, NDCG = \frac{1}{N} \sum_{u} \frac{DCG(u)}{DCG^*(u)} \tag{3}$$

Precision and recall were combined using F_1 score (Eq. (4)). The precision of a user u is the percentage of good predictions among all the predictions made for user u. The recall is the percentage of good predictions among the amount of pairs of items in the current iteration. Final precision and recall of the test set are obtained by considering the harmonic mean of average precision and average recall of each user.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

Besides those metrics, we further analyze how the user ratings profile length and the number of friends impact the recommendation quality through two other metrics: (1) profile length factor and (2) social degree:

Profile length Factor: Let $\bar{R}$ be an average number of user ratings and an α coefficient, where $\alpha \in \mathbb{R}$. Eq. (5) represents the profile length factor calculus. In our experiments (Figure 6(a)) we compute the F_1 score for different profile length factors to determine the number of ratings necessary to better select a recommendation model for a given dataset.

$$Pl_{factor} = \alpha * \bar{R} \tag{5}$$

Social Degree: The social degree is given by the average degree of the social network ($\bar{S}$) and a β coefficient where $\beta \in \mathbb{R}$. We compute the social degree according to Eq. (6). Using different number of friends to select a recommendation model we evaluate the F_1 results (Figure 6(b) - 6(f)).

$$S_d = \beta * \bar{S} \tag{6}$$

5. RESULTS

In this section, we thoroughly assess the effectiveness of our proposed pairwise preference recommender approach, Social PrefRec. First, we analyze the quality of recommendations on the datasets (*Q1*). Then, we measure the relevance of recommendations (*Q2*), focusing on the ranking relevance of Social PrefRec compared to those provided by three social recommender systems, besides the original PrefRec. Furthermore, we measure the performance for each social metric (*Q3*) and under different sparsity levels (*Q4*). We close this section by analyzing how user's profile length versus its social degree (*Q5*) and popular versus unpopular users (*Q6*) influence the quality of the recommendations.

5.1 How accurately social information help on pairwise preference recommendation? (*Q1*)

F_1 scores are represented in Figure 4, for minimum threshold and average connection weight selection model strategies. Against all datasets with a profile length of 30-ratings for PrefRec versus 0-ratings for Social PrefRec, the social approach achieved better results using *Minimum threshold* strategy. Rate-15-items baseline is widely used to bootstrap traditional recommender systems [Chang et al. 2015]. Thus, to make a fair comparison we give 30-ratings for PrefRec, which means that all runs have a good safe margin and should not harm its performance. Nevertheless, our social approach performs at least equivalently to traditional one, as we further discuss on *Q3*. Note that those results are on cold-start scenarios: under scenarios where a user provides enough ratings, a social approach does not add much value.

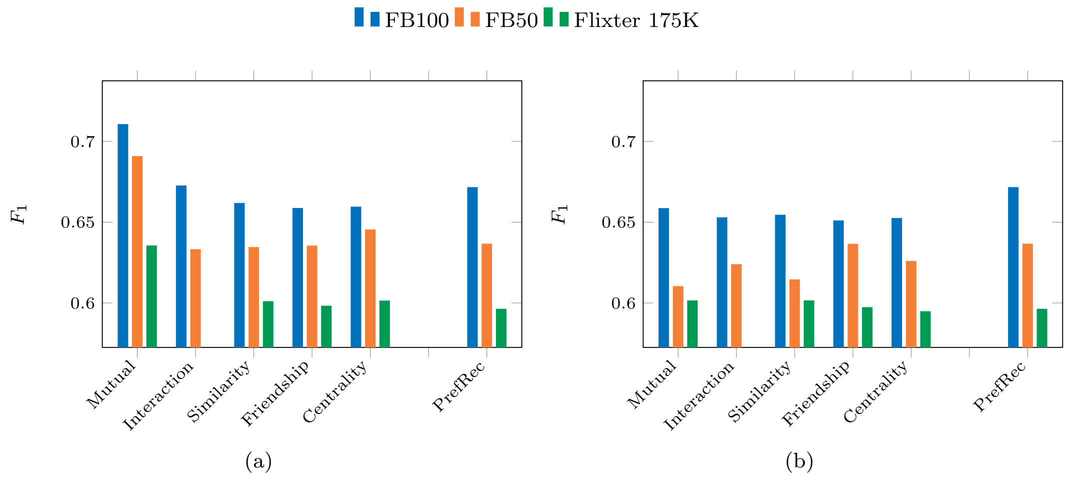

Fig. 4. F_1 scores for Social PrefRec and PrefRec for 2 model selection strategies: (a) *Minimum threshold* and (b) Average connection weight.

5.2 How relevant are the recommendations made by a social pairwise preference recommender? (*Q2*)

Tables VI, VII, VIII, and IX show the *nDCG* results for rank size 5, 10, 15, 20, under *minimum threshold* strategy, Section 3.2, and 0-rating scenario. We apply each approach described in Section 4.3 on each dataset to assess the robustness of each. We observe that Social PrefRec obtains better results for new users compared to the other social recommenders. One of the main reasons for the effective performance of our approach is that it chooses a suitable recommendation model based on a consensual set of friends' preferences. The other approaches not only consider all friends' preferences, but SocialMF, for example, also relies on trust propagation mechanism, which incorporates preferences from friends of friends. Thus, we argue that a specific set of friends (neighbors) might be a better source to give more relevant recommendations.

Another main difference is about how each approach deals with item attributes. Matrix factorization profiles both users and items in a user-item rating matrix and through latent factor models project items and users into the same latent space, thus making them comparable.

According to a Kruskal-Wallis test with 95% confidence, Social PrefRec performance is significantly better than social matrix factorization approaches. Mutual Friends is better than others Social PrefRec metrics in ndcg@5. For ndcg@10, there is no significant difference between Mutual Friends, Centrality, Friendship, Similarity and Interaction. The performance with Centrality achieves an equivalent score as Mutual Friends in ndcg@15 results. Finally, the ndcg@20 values show that Mutual Friends, Centrality, Friendship and Similarity are not significantly different.

Table VI. Resulting nDCG@5, @10, @15, and @20 against FB50.

Approach	Size of Rank			
	@5	@10	@15	@20
SoRec	0.8515 ± .138	0.8412 ± .123	0.8340 ± .114	0.8297 ± .108
SocialMF	0.7469 ± .183	0.7536 ± .158	0.7550 ± .146	0.7576 ± .139
TrustMF	0.8373 ± .147	0.8296 ± .133	0.8259 ± .122	0.8250 ± .114
Friendship	0.9870 ± .035	0.9779 ± .039	0.9697 ± .040	0.9612 ± .042
Similarity	0.9860 ± .036	0.9770 ± .040	0.9683 ± .042	0.9601 ± .045
Centrality	0.9881 ± .033	0.9802 ± .038	0.9721 ± .039	0.9647 ± .041
Mutual	**0.9934** ± .025	**0.9890** ± .028	**0.9752** ± .033	**0.9665** ± .038
Interaction	0.9822 ± .043	0.9733 ± .046	0.9661 ± .047	0.9589 ± .046

Table VII. Resulting nDCG@5, @10, @15, and @20 against FB100.

Approach	Size of Rank			
	@5	@10	@15	@20
SoRec	0.8358 ± .141	0.8251 ± .124	0.8180 ± .119	0.8114 ± .115
SocialMF	0.7124 ± .193	0.7100 ± .173	0.7111 ± .163	0.7166 ± .155
TrustMF	0.7742 ± .149	0.7819 ± .128	0.7835 ± .120	0.7804 ± .115
Friendship	0.9852± .036	0.9746 ± .042	0.9666 ± .044	0.9582 ± .046
Similarity	0.9850 ± .038	0.9746 ± .042	0.9667 ± .043	0.9587 ± .046
Centrality	0.9897 ± .028	0.9797 ± .037	0.9706 ± .041	0.9621 ± .044
Mutual	**0.9933** ± .023	**0.9836** ± .027	**0.9715** ± .037	**0.9636** ± .042
Interaction	0.9762 ± .053	0.9762 ± .060	0.9603 ± .061	0.9547 ± .061

Table VIII. Resulting nDCG@5, @10, @15, and @20 against Flixter 175K.

Approach	Size of Rank			
	@5	@10	@15	@20
SoRec	0.8209 ± .134	0.8236 ± .120	0.8224 ± .115	0.8214 ± .111
SocialMF	0.7715 ± .138	0.7753 ± .126	0.7755 ± .123	0.7751 ± .120
TrustMF	0.7603 ± .136	0.7521 ± .127	0.7494 ± .123	0.7485 ± .120
Frienship	0.9840 ± .039	0.9769 ± .038	0.9713 ± .038	0.9671 ± .039
Similarity	0.9852 ± .038	0.9779 ± .037	0.9726 ± .037	0.9675 ± .039
Centrality	0.9830 ± .039	0.9758 ± .039	0.9704 ± .038	0.9657 ± .040
Mutual	**0.9916** ± .023	**0.9810** ± .030	**0.9772** ± .032	**0.9766** ± .030

Table IX. Resulting nDCG@5, @10, @15, and @20 against Flixter 811K.

Approach	Size of Rank			
	@5	@10	@15	@20
SoRec	0.8198 ± .131	0.8173 ± .118	0.8145 ± .113	0.8133 ± .109
SocialMF	0.7279 ± .139	0.7335 ± .122	0.7359 ± .116	0.7374 ± .113
TrustMF	0.7204 ± .135	0.7246 ± .122	0.7246 ± .117	0.7298 ± .113
Frienship	0.9810 ± .044	0.9748 ± .044	0.9699 ± .043	0.9662 ± .042
Similarity	0.9804 ± .045	0.9744 ± .044	0.9696 ± .044	0.9661 ± .043
Centrality	0.9809 ± .044	0.9742 ± .044	0.9699 ± .043	0.9667 ± .042
Mutual	**0.9908** ± .029	**0.9812** ± .036	**0.9747** ± .038	**0.9685** ± .041

5.3 Which social metrics are more important for item recommendation? (*Q3*)

We perform Kruskal-Wallis test to check statistical significance among Social PrefRec metrics results and PrefRec, see Figure 4(b). Mutual Friends, Interaction, Similarity are indicated as best performing. Furthermore, Friendship and Centrality results are not significantly different from PrefRec (profile length = 30-ratings) result. Thus, the test shows with 95% confidence, that with the first three metrics we can better recommend in social 0-rating profile scenario than 30-rating profile in a traditional recommender approach. Although the others social metrics achieved the same result as the traditional approach, they need none previous rating from a user.

5.4 How effective is Social PrefRec to mitigate data sparsity problems? (*Q4*)

As sparsity is a big challenge faced by recommendation systems, we consider five subsets sampled from FB100. The basic idea is to simulate sparse scenarios where input datasets has many items to be rated with very few/sparse ratings per user. For instance, $FB100_{50}$ was obtained by eliminating around 50% of the ratings in FB100 in a stratified way, so we keep homogeneous subgroups of the original set. Table X shows the characteristics of the datasets extracted from FB100.

Table X. FB100 sparse subsets.

$FB100$ ($Dataset$)	Ratings per user (Average)	Sparsity (%)
10	137.9	18.4
20	122.6	27.4
30	107.28	36.6
40	91.8	45.7
50	76.3	54.8

Figure 5 shows that PrefRec is superior on less sparse datasets. However, the social approaches on sparser dataset, i.e. $FB100_{50}$ and $FB100_{40}$, exhibit better recommendations quality, particularly for Mutual connection weight metric. These results complement previous analyses of Social PrefRec.

5.5 Does social degree affect Social PrefRec as much as profile length affects PrefRec? (*Q5*)

Traditional recommender systems present better performance when they know more user's preferences. Figure 6(a) shows the prediction performance of PrefRec on two Facebook datasets. We observe that the recommender predictions get better as the user's profile gets longer. For instance, PrefRec achieves F_1 equal to 71.18 on FB100 when we use 123 ratings for recommendation model selection ($\alpha = 0.8$).

However, with Social PrefRec, we do not note a correlation between social degree and prediction performance. Figures 6(b) to 6(f) show the results for different social degrees. The overall picture is the same on all datasets and all social metrics. So, increasing the number of friends to select a recommendation model do not increase the F_1 score. This leads us to the next question that further evaluates all social metrics for higher and lower social degrees.

5.6 Are there major differences between the quality of recommendations considering popular and unpopular users? (*Q6*)

To investigate the effects of social degree on Social PrefRec, we begin by recalling the definition of popular and unpopular users. First, we calculate the average number of friends on the subset $FB50$ and $FB100$. Popular users are those that have more than the average number of friends, whereas unpopular users have only half the average number of friends.

Figure 7 shows the (F_1) achieved by Social PrefRec for each social metric against each subset.

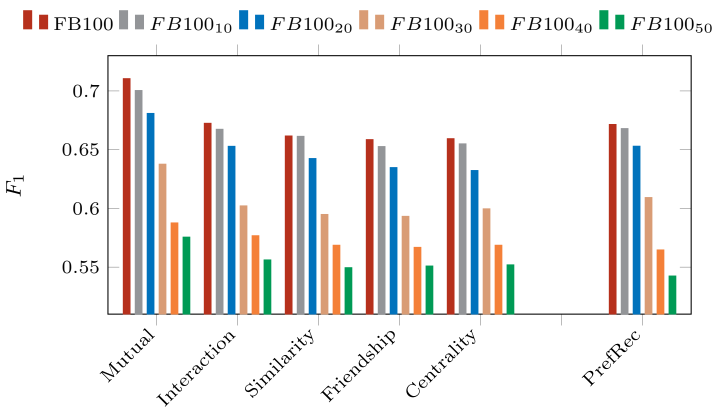

Fig. 5. Social PrefRec and PrefRec metrics across sparse scenarios with *minimum threshold* of 40%.

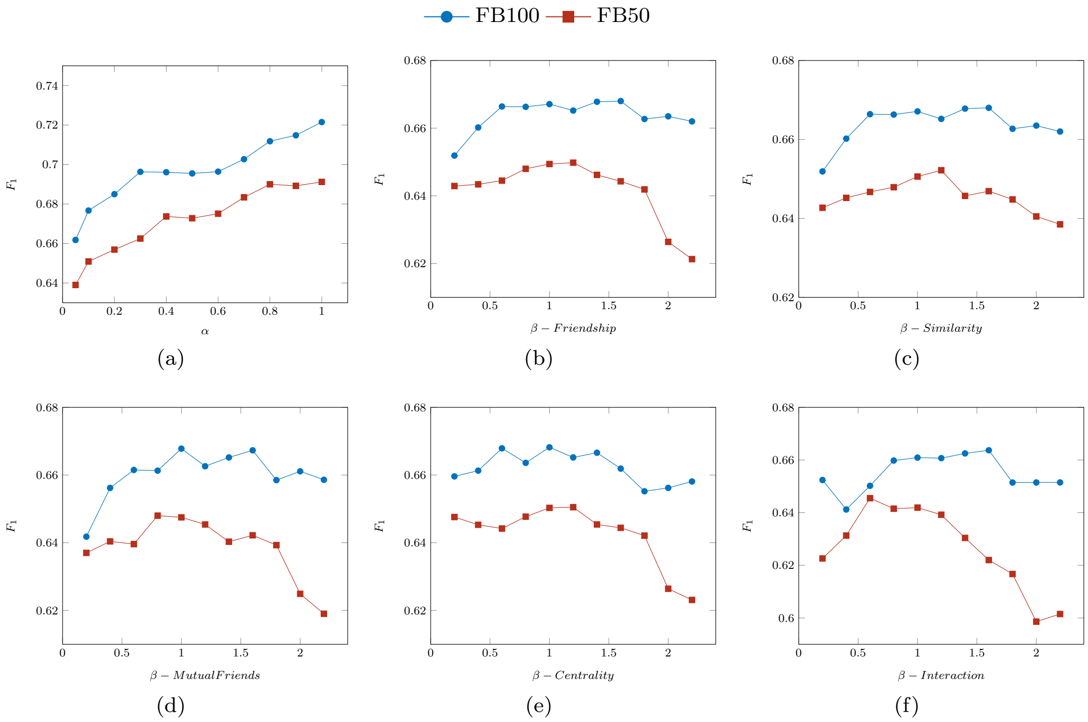

Fig. 6. Profile length factor effect (α, see Eq. 5) over F_1 measure (PrefRec) in Fig. 6(a). Social degree effect variants (β, see Eq. 6) over F_1 measure in Fig. 6(a) – 6(f).

Note that, the overall performance is similar between each subset. Regarding the major differences between popular and unpopular users, the mutual friends social metric achieves the worst results, which shows the need of larger amounts of friends to better select a recommendation model. On the other hand, the centrality social metric performance shows that it is not affected by the number of friends.

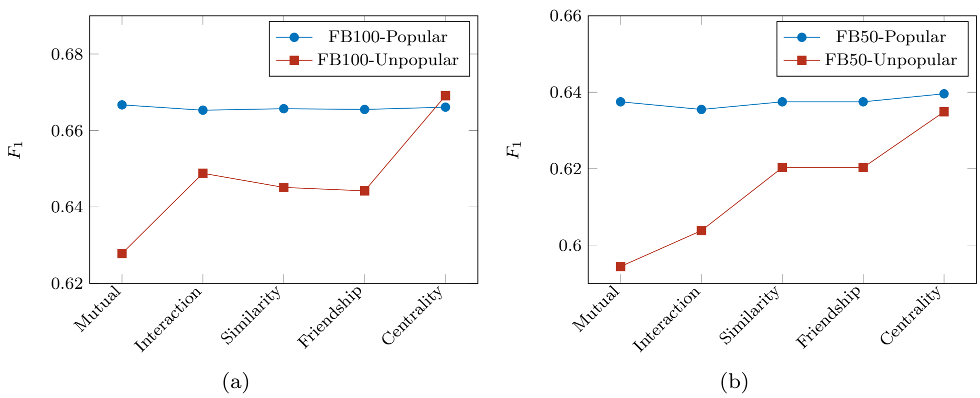

Fig. 7. (a) F_1 metric for Popular users and Unpopular users in FB100 and (b) FB50 with *minimum threshold* of 10%.

6. CONCLUSION

We have devised and evaluated Social PrefRec, an approach whose goal is to help pairwise preferences recommender systems to deal with 0-rating user's profile. Driven by six research questions, we expand earlier work by analyzing and demonstrating the effectiveness of our proposed social preference learning approach. Our analyses were performed on four real datasets. We also carefully investigate the role of five well-known social metrics in pairwise preference recommendation and proposed a clustering based approach to incorporate social networks into recommender systems. With Social PrefRec approach, we brought novel ways to extend traditional recommenders.

Finally, although focused on social networks, our work could be extended to tackle other networks (graphs) where we can compute similarity scores between nodes, such as scientific networks or inferred networks [Felício et al. 2016]. Another interesting direction for future work is the study of how to choose more influential nodes, *e.g.* find out the friends who have a stronger influence on a user and apply their preferences to tackle cold-start recommendations.

REFERENCES

Alexandridis, G., Siolas, G., and Stafylopatis, A. Improving Social Recommendations by Applying a Personalized Item Clustering Policy. In *Proceedings of the ACM Conference on Recommender Systems*. Hong Kong, China, pp. 1–7, 2013.

Balakrishnan, S. and Chopra, S. Two of a Kind or the Ratings Game? Adaptive Pairwise Preferences and Latent Factor Models. *Frontiers of Computer Science* 6 (2): 197–208, 2012.

Canamares, R. and Castells, P. Exploring Social Network Effects on Popularity Biases in Recommender Systems. In *Proceedings of the ACM Conference on Recommender Systems*. Foster City, CA, USA, pp. 1–8, 2014.

Chang, S., Harper, F. M., and Terveen, L. Using Groups of Items for Preference Elicitation in Recommender Systems. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing*. Vancouver, BC, Canada, pp. 1258–1269, 2015.

Cohen, W. W., Schapire, R. E., and Singer, Y. Learning to Order Things. *Journal of Artificial Intelligence Research* 10 (1): 243–270, 1999.

de Amo, S., Bueno, M. L. P., Alves, G., and da Silva, N. F. F. Mining User Contextual Preferences. *Journal of Information and Data Management* 4 (1): 37–46, 2013.

de Amo, S. and Oliveira, C. G. Towards a Tunable Framework for Recommendation Systems Based on Pairwise Preference Mining Algorithms. In *Proceedings of the Canadian Conference on Artificial Intelligence*. Montreal, Canada, pp. 282–288, 2014.

de Amo, S. and Ramos, J. Improving Pairwise Preference Mining Algorithms using Preference Degrees. In *Proceedings of the Brazilian Symposium on Databases*. Curitiba, Brazil, pp. 107–116, 2014.

Felício, C. Z., de Almeida, C. M. M., Alves, G., Pereira, F. S. F., Paixão, K. V. R., and de Amo, S. Visual Perception Similarities to Improve the Quality of User Cold Start Recommendations. In *Proceedings of the Canadian Conference on Artificial Intelligence*. Victoria, Canada, pp. 96–101, 2016.

Felício, C. Z., Paixão, K. V. R., Alves, G., and de Amo, S. Social PrefRec framework: leveraging Recommender Systems Based on Social Information. In *Proceedings of the Symposium on Knowledge Discovery, Mining and Learning*. Petrópois, RJ, Brazil, pp. 66–73, 2015.

Guo, G., Zhang, J., Sun, Z., and Yorke-Smith, N. LibRec: a Java Library for Recommender Systems. In *Proceedings of the Conference on User Modeling, Adaptation, and Personalization*. pp. 1–4, 2015.

Jamali, M. and Ester, M. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. In *Proceedings of the ACM Conference on Recommender Systems*. Barcelona, Spain, pp. 135–142, 2010.

Krohn-Grimberghe, A., Drumond, L., Freudenthaler, C., and Schmidt-Thieme, L. Multi-relational Matrix Factorization Using Bayesian Personalized Ranking for Social Network Data. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. Seattle, Washington, USA, pp. 173–182, 2012.

Ma, H., Yang, H., Lyu, M. R., and King, I. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Proceedings of the ACM Conference on Information and Knowledge Management*. Napa Valley, CA, USA, pp. 931–940, 2008.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender Systems with Social Regularization. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. Hong Kong, China, pp. 287–296, 2011.

Ma, H., Zhou, T. C., Lyu, M. R., and King, I. Improving Recommender Systems by Incorporating Social Contextual Information. *ACM Transactions on Information Systems* 29 (2): 9:1–9:23, 2011.

Nielsen, T. D. and Jensen, F. V. *Bayesian Networks and Decision Graphs*. Springer Science & Business Media, 2009.

PARK, S.-T. AND CHU, W. Pairwise Preference Regression for Cold-start Recommendation. In *Proceedings of the ACM Conference on Recommender Systems*. New York, NY, USA, pp. 21–28, 2009.

SAMMUT, C. AND WEBB, G. I. Leave-One-Out Cross-Validation. In *Encyclopedia of Machine Learning*. Springer, Boston, MA, USA, pp. 600–601, 2010.

SHANI, G. AND GUNAWARDANA, A. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor (Eds.). Springer, Boston, MA, USA, pp. 257–297, 2011.

SHARMA, A. AND YAN, B. Pairwise Learning in Recommendation: Experiments with Community Recommendation on Linkedin. In *Proceedings of the ACM Conference on Recommender Systems*. Hong Kong, China, pp. 193–200, 2013.

TANG, J., HU, X., AND LIU, H. Social Recommendation: a Review. *Social Network Analysis and Mining* 3 (4): 1113–1133, 2013.

YANG, B., LEI, Y., LIU, D., AND LIU, J. Social Collaborative Filtering by Trust. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Beijing, China, pp. 2747–2753, 2013.

YUAN, T., CHENG, J., ZHANG, X., LIU, Q., AND LU, H. How Friends Affect User Behaviors? An Exploration of Social Relation Analysis for Recommendation. *Knowledge-Based Systems* vol. 88, pp. 70–84, 2015.