Preference-like Score to Cope with Cold-Start User in Recommender Systems

Crícia Z. Felício*‡, Klérisson V. R. Paixão‡, Celia A. Z. Barcelos‡, Philippe Preux†
*Federal Institute of Triângulo Mineiro, IFTM
‡Federal University of Uberlândia, UFU
Uberlândia (MG), Brazil
E-mail: cricia@iftm.edu.br, {klerisson, celiazb}@ufu.br
†University of Lille & CRIStAL
Villeneuve d'Ascq, France
E-mail: philippe.preux@inria.fr

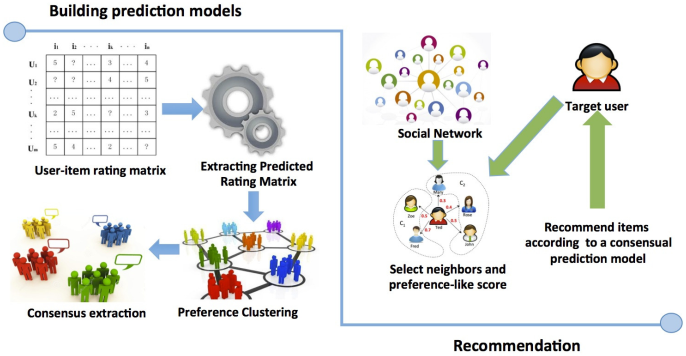
***Abstract*—In recent years, there has been an explosion of social recommender systems (SRS) research. However, the dominant trend of these studies has been towards designing new prediction models. The typical approach is to use social information to build those models for each new user. Due to the inherent complexity of this prediction process, for full cold-start user in particular, the performance of most SRS fall a great deal. We, rather, propose that new users are best served by models already built in system. Selecting a prediction model from a set of strong linked users might offer better results than building a personalized model for full cold-start users. We contribute to this line of work comparing several matrix factorization based SRS under full cold-start user scenario; and proposing a general model selection approach, called ToSocialRec, that leverages existing recommendation models to offer items for new users. Our framework is not only able to handle several social network connection weight metrics, but any metric that can be correlated with preference similarity among users, named here as Preference-like score. We perform experiments on real life datasets that show this technique is as efficient or more than current state-of-the-art techniques for cold-start user. Our framework has also been designed to be easily deployed and leveraged by developers to help create a new wave of SRS.**

***Index Terms*—New user experience; New user problem; Recommender Systems; Cold-start User; Social Recommender**

I. Introduction

Research on Social Recommender Systems (SRS) has exploded in the past decade. This is mainly because social media content now accounts for most data published on the Web. With such an abundance of information, SRS can infer user's preference from social contexts to offer more accurate recommendations. This is of particular interest for a new user (also known as a *cold* user), because he is considered initially by the recommendation system though he has not yet provided any information about his preferences.

There has been substantial research interest in improving certain aspects of the user cold-start problem [1]–[6]. However, the dominant trend of these studies has been towards designing new prediction models. The typical approach is to use social information to build a recommendation model for each new user. Our own work has followed this standard path. Due to the inherent complexity of this modeling process, the performance of most SRS decreases for cold users: those systems cannot offer personalized recommendations until they collect enough preference information from users.


Building prediction models
User-item rating matrix
Extracting Predicted Rating Matrix
Preference Clustering
Consensus extraction
Social Network
Target user
Select neighbors and preference-like score
Recommend items according to a consensual prediction model
Recommendation

	i1	i2	· · ·	ik	· · ·	in
U1	5	?	. . .	3	. . .	4
U2	?	?	. . .	4	. . .	5
·	. . .	. . .	. . .	. . .	. . .	. . .
Uk	2	5	. . .	?	. . .	3
·	. . .	. . .	. . .	. . .	. . .	. . .
Um	5	4	. . .	2	. . .	?

Fig. 1. ToSocialRec selects a prediction model from a set of consensual ones previously built for other users.

This paper is our reaction to these experiences. Figure 1 illustrates the method we propose. ToSocialRec takes advantage of all prediction models already built in the system. It chooses the most suitable one and creates a consensual model for a new user based on how strong he is similar to the other users. The selection process is based on the *homophily* principle, which is the tendency of individuals to associate and bond with similar others.

Earlier social approaches investigated the homophily assumption to build new prediction models. Herein, we focus on inspecting the existing models, looking for those that might maximize information gained about the new user.

Our hypothesis is that *selecting a prediction model from a set of strongly similar users would offer better or equivalent results than building a personalized model for cold users*. We structure our work around the following research question:

RQ: Does a recommender system already hold suitable prediction models to deal with a cold user?

Model selection is a broad subject. We will look at two distinct sub-questions to examine ToSocialRec:

RQ#1: How well can a selected model predict the ratings of new users?

RQ#2: How well can a selected model rank items to new users?

Our main contributions are threefold:

1) We compare and contrast several matrix factorization based social recommender systems in a cold-start scenario;
2) We propose a general model selection approach that leverages existing models to recommend relevant items to cold users;
3) We thoroughly evaluate ToSocialRec against six distinct data sets and we show its effectiveness in contrast to state-of-the-art matrix factorization based recommenders.

This paper is organized as follows. First, we further motivate the need for a new SRS and its main concept (Section II). Next we present the framework and how it works (Section III). Section IV describes our experimental settings and results. Then, Section V discusses related work and Section VI concludes the paper.

II. Background

In this section, we present an illustrative example of ToSocialRec and introduce the main concepts underlying this work.

A. Motivating example

As a motivating example, let us consider the domain of movie recommendation. Let us assume that the recommender system has only two different genres of movies: romance and thriller. The system has yet dealt with people who prefer romances and others fond of thrillers. We can find these groups based on the ratings already given. The first group will have users that have given high ratings to romantic movies and the second group will be made of people who give high ratings to thrillers. Now, let us assume a new (cold) user u looking for help to find movies he would like to watch. A common situation is when the new user has yet given no rating; he is admitted into the system in exchange of his social information. Once the system has collected enough preference data from the user, it can build an initial prediction model. Later, as the user provides ratings, the system improves the model. For now, the recommender system may associate the cold user with other users. For instance, it might associate u along with users from the same gender, age or present similar affinities. Assuming that u has a lot in common with people who like romances rather than thrillers, we might further inspect prediction models in the romance group to select one for u.

We hypothesize that it is reasonable to use people's preferences from one of identified groups to offer a recommendation to a new user. To select one group, we use features that characterize the connection strength between connected users.

B. Preference-like Score

Preference-like score is the information that can be correlated with preference similarity among users. Through this score, we can filter a set of users whose prediction models may be a good enough for initial recommendations.

We assume the existence of a network among users to compute the connection weight between them. For instance, preference-like score could be the demographic similarity in a friendship network or the centrality degree.

Formally, we can represent the users network as a graph $G = (V, E)$, in which users are the vertices of this graph. A set of friends (neighbors) of a vertex u is $F(u) = \{v | v \in V \wedge (u, v) \in E\}$ and a function $l : F \rightarrow \mathbb{R}$ defines *Preference-like Score* between u and v in $[0, 1]$.

Social network presents several ways to compute the preference-like score. In this paper, we exploit the following well-known network metrics [7], [8] as defined in [9], [10]:

- **Friendship:** This score is equal to 1 for each connection between the target user and a neighbor.
- **Mutual Friends:** This is given by the mutual neighbors score computed by Jaccard coefficient;
- **Similarity:** This is the demographic similarity between a target user and his neighborhood;
- **Centrality:** We can set the connection weight according to centrality of the target user neighbors in the social network.

We advocate that ToSocialRec is not restricted to the commonly used social network metrics. Therefore, we also analyze the performance of the preference-like score given by non traditional networks similarities, using visual perception networks. Based on our earlier work [11], we define the following metrics:

- **VP-similarity:** Similarity score based on visual perception's similarities among users;
- **VP-friendship:** A specialized friendship connection represented by visual perception network where users in the same visual perception cluster are connected and have preference-like score equal to 1.

In summary, we generalize the concept of network similarity and that is why we call it ***Preference-like score***. We argue that if we can define a function l that determines the connection strength between users in a recommender system, we can use this score to help a recommender system dealing with new users.

III. ToSocialRec

In this section, we describe ToSocialRec highlighting how to incorporate the preference-like score in Matrix Factorization based recommender systems.

Let U be a set of users and I be a set of items. Each user $u \in U$ and each item $i \in I$ has a unique identifier. The user-item rating matrix is $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ is the rating given by user u on item i, and m is the number of users, and n is the number of items. An example of a user-item rating matrix with 6 users and 7 item, and ratings in the range $\{1, 2, 3, 4, 5\}$ is provided in table I.

In traditional recommender systems, the recommendation task is based on the prediction of the missing values of the user-item rating matrix. Then, predictions are used to rank items and recommend the k top-ranked.

In our work, we present ToSocialRec, an approach to extend traditional recommender systems to incorporate preference-like scores. The goal is to deal with cold-start users. ToSocialPrec alternates two phases, (a) construction and update of the prediction models, and (b) making recommendations. These two phases are described in the next two subsections.

A. Construction and Update of the Prediction Models

The main steps of the prediction model construction are: (i) Ratings prediction, (ii) Preference clustering, and (iii) Consensus computation. Each step is detailed below. To keep track of the evolving nature of the environment (such as the set of available ratings), the model has to be updated; though essential in a live system, this step is not considered in this paper and only briefly described below: indeed, this step is beyond the scope of this paper.

Rating prediction: from the user-item rating matrix, we use a matrix factorization technique to get a matrix of predicted ratings R'. R' is expressed as a product of latent factors, $R' = PQ^T$, where P is the user latent factor matrix, and Q is the item latent factor matrix. The predicted rating of the item i_k by user u_j is $R'_{u_j,i_k} = predict(u_j, i_k, P, Q)$, the details of this function depending on the completion method being used.

As an example, Table II shows the predicted rating matrix R' obtained from the user-item matrix of table I, as completed using the BiasedMF[1] algorithm [12]. With BiasedMF, the prediction function is $predict(u_j, i_k, P, Q) = \mu + b_{u_j} + b_{i_k} + P_{u_j} Q^T_{i_k}$, where μ is the overall average rating, b_{u_j} is the deviation from μ of user u_j ratings, b_{i_k} is the deviation from μ of item i_k ratings, P_{u_j} is the u_j^{th} row of matrix P which are the latent factors for user u_j, and $Q_{i_k}^{\text{th}}$ row of matrix Q which are the latent factors for item i_k. Finally, given the predicted rating matrix R', the preference vector for a user u_j is defined as the predicted ratings for user u_j, $\theta_j = R'_{u_j}$.

Preference clustering: Given a predicted rating matrix R', we can cluster users according to their preference vectors, that is the rows of R'. A distance function and a clustering algorithm $\mathcal{C}$ are used. After clustering, we have a set of cluster C, where each cluster C_s contains a set of users with the similar preferences.

Consensus computation: for each cluster C_s, we apply a consensus operator $\mathcal{A}$ to get the consensual preference vector $\hat{\theta}_s$ of cluster C_s. In this paper, the operator is the average, that is $\hat{\theta}_{s,k}$ is the average predicted rating for item k. We obtain $M = \{M_1 = (C_1, \hat{\theta}_1), \ldots, M_K = (C_K, \hat{\theta}_K)\}$, the set of prediction models where each M_s is composed of a cluster of users C_s and its consensual preference vector $\hat{\theta}_s$.

TABLE II
PREDICTED RATING MATRIX.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
Zoe	4.6	2.09	4.23	4.24	4.84	1.07	1.0
Fred	4.2	3.8	4.42	5.0	4.86	2.28	1.2
Mary	1.97	4.84	3.22	4.87	2.68	2.68	1.61
Rose	1.19	3.24	2.17	3.56	1.92	1.23	1.0
Paul	1.77	3.16	2.81	4.07	1.14	1.6	1.56
John	2.09	4.32	3.29	4.77	2.09	2.46	1.98

Table III continues the example and exemplifies the clustering process, and the consensus computation: the users from Table II were clustered in two groups according to their preference vectors, and the consensual preference vector for each cluster was computed.

Model Update: in a live recommendation system, the set of prediction models M must be rebuilt when the insertion of new ratings in the rating matrix R increases the difference between R and prediction rating matrix R'. Let $D(t) = [d_{u,i}]_{m \times n}$ be the absolute difference matrix between R and R' at time t, where each $d_{u,i} = |r_{u,i} - r'_{u,i}|$. Let us define a difference score $diff(t) = \sum_{j=1}^{m} \sum_{k=1}^{n} d_{u_j,i_k}$. After each update on R at any time $t' > t$, it is straightforward to update $diff(t')$ incrementally. Then, we decide on updating M once $diff(t')$ reaches a certain threshold.

B. Making Recommendations

In its second phase, ToSocialRec makes use of a prediction model M_s to recommend items for a **new user**. The recommendation process is executed online, differently from the previous phase which is offline.

The selection of a prediction model uses preference-like scores. Let $select : U \to M$ be a function that selects the suitable prediction model from M for a target user u defined by the minimum threshold strategy, adapted from our previous works [9], [10], as follows.

Minimum threshold: let $\varepsilon \in [0, 1]$ be a preference-like minimum threshold. The minimum threshold strategy selects the prediction model $M_s \in M$ which associated cluster of

TABLE I
EXAMPLE OF A USER-ITEM RATING MATRIX. - MEANS THAT THE USER HAS NOT RATE THE ITEM.

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

TABLE III
CONSENSUAL PREFERENCE VECTORS.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
Zoe	4.6	2.09	4.23	4.24	4.84	1.07	1.0
Fred	4.2	3.8	4.42	5.0	4.86	2.28	1.2
$\hat{\theta}_1$	4.4	2.94	4.32	4.62	4.85	1.67	1.1
Mary	1.97	4.84	3.22	4.87	2.68	2.68	1.61
Rose	1.19	3.24	2.17	3.56	1.92	1.23	1.0
Paul	1.77	3.16	2.81	4.07	1.14	1.6	1.56
John	2.09	4.32	3.29	4.77	2.09	2.46	1.98
$\hat{\theta}_2$	1.76	3.89	1.98	4.32	2.87	1.99	1.53

[1] The name BiasedMF comes from the LibRec library that we use in the experiments.

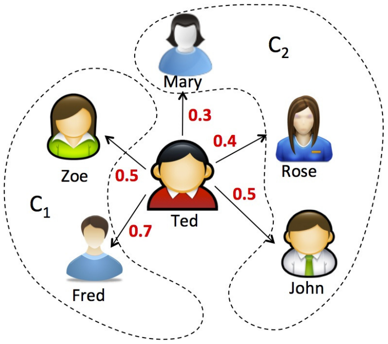


Fig. 2. Example of a preference-like network with a cold-start user (Ted) and his neighbors. Dashed contours identify the two preference clusters.

users C_s that has more friends of u satisfying a threshold, according to the Eq. (1).

$$select(u) = \arg\max_{M_S \in M} |\{v \in F(u) \wedge l(u,v) \geq \varepsilon\}| \quad (1)$$

The recommendation process for a cold-start user is executed as follows:

1) Given a target user u and a Preference-like metric, the system will select the neighbors $F(u)$ of u, and the Preference-like score, previously computed, between u and each $v \in F(u)$.
2) Using minimum thresholding, select the prediction model M_s according to eq. (1).
3) The consensual preference vector of M_s $\hat{\theta}_s$, is used to rank the items.
4) k top-ranked items are recommended to u.

Example. To explain how our recommendation phase works, we consider the preference-like network in Figure 2. The user Ted (our cold-start user) is connected with users of the two computed groups (see Table III). Preference-like score between Ted and his connections stands for the level of similarity between them. Given a minimum preference-like score of 0.5, we identify that C_1 is the group with more users satisfying this threshold. We will use the consensual predictions of group C_1, $\hat{\theta}_1$ to offer recommendations to Ted. Then, the item ranking is: $\{i_5, i_4, i_1, i_3, i_2, i_6, i_7\}$.

IV. Experiments and Discussions

In this section, we conduct extensive experiments on several datasets. We also compare ToSocialRec against a set of state of the art algorithms. Then, we evaluate how statistically significant our results are.

A. Datasets and Configuration

We evaluate ToSocialRec on six datasets: 3 movie rating datasets, 1 product review dataset, 1 painting dataset, and 1 clothing dataset. Table IV summarizes their main descriptive features. It is noteworthy that those datasets present a variety of features among themselves. We briefly describe each dataset:

Facebook Dataset [9] was collected by ourselves with a Facebook web application we developed. This dataset has personal information: relationship status, age bracket, gender, born-in, lives-in, religion and study-in, and each volunteer's friendship network was crawled: friend's relationships, mutual friends, and friends' centrality.

Flixster Dataset [13] contains social information. It 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. Note that we consider only a fraction of the original Flixster dataset. We further discuss this point in Section IV-E.

Filmtrust Dataset [14] is about movie sharing and ratings. The preference-like network is based on users trust network. From a trust network, we compute a mutual friend score, and a user centrality score.

Epinions Dataset [15] contains data from product reviews. Epinions' preference-like network is also computed with users trust network. From trust network we compute the mutual friend score, and the user centrality score. Due to computational restrictions, we only consider a fraction of the original dataset. We further discuss this point in Section IV-E.

Paintings Dataset [11] contains information about how a set of users look at a set of paintings. Users may be clustered based on this information, and a preference-like network is built based on this clustering. A similarity score between users is computed using the distance between visual perception of pair of users belonging to the same cluster. Visual perception information was collected through an eye tracker device.

Clothing Dataset [16] similarly to the Paintings dataset, visual perception information was collected through an eye tracker device while users were looking at clothes. Users are clustered and a preference-like network is inferred. From the original dataset we got only female clothing subset and items rated in common among all users.

B. Evaluation Metrics

As we focus on cold-start users, we adopt the leave-one-out protocol [17]: at each round, we train on all users but one who is used as a test user. As we do not use any information about the test user, it is a cold-start user. We call this protocol as **0-ratings protocol**.

The goal is to measure the performance of each algorithm to predict item ratings. We use the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), Equation (2), as evaluation criterion, where $r_{u,i}$ is the rating for an item i from a target user u, $\hat{r}_{u,i}$ is the predicted rating for i and X is the total number of ratings.

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \hat{r}_{u,i}|}{X}, RMSE = \sqrt{\frac{\sum_{u,i}(r_{u,i} - \hat{r}_{u,i})^2}{X}} \quad (2)$$

Though widely used, MAE and RMSE do not characterize the quality of the recommendation. Ranking quality is measured computing the Normalized Discounted Cumulative Gain (NDCG) metric, Equation (3). In that equation, $r_{u,1}$ is the rating (according to the ground truth) of the item at the first

TABLE IV
DATASET FEATURES.

Dataset	Users	Items	Ratings	Sparsity (%)	Ratings / User (Average)	Links	Links / User (Average)
Facebook	498	169	49,729	40.9	99.85	5,468	10.9
Flixster	1,323	1,175	811,726	47.78	613.54	6,526	5.34
FilmTrust	1,508	2,071	35,494	98.86	23.53	1,853	3.0
Epinions	1,161	529	25,781	95.8	22.2	62,903	55.03
Paintings	194	605	38,753	67	200	28,992	149.44
Clothing	121	210	25,396	0.05	209.88	7,204	59.53

ranking position, p is the ranking position, $r_{u,p}$ is the ground truth rating for the item in the rank position p and M is the size of the ranked list. $DCG(u)$ 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.

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

C. Other methods

To assess the effectiveness of ToSocialRec, we compare it with other social matrix factorization based recommender systems. These methods were designed to combine social information with rating data. They are distinct from ToSocialRec that uses social information only to select a consensual prediction model between preference clusters. The weight of social information in the building model process is determined by a parameter in the three approaches. None of them makes use of any clustering technique.

SoRec [18] is based on latent factors of items, users, and social network relationships. The influence of one neighbor in a rating prediction increases if he is trusted by a lot of users and decreases if the target user has many connections.

SocialMF [13]: applies a trust propagation mechanism. More distant users have less influence (weight) in rating prediction than the trusted direct contacts.

TrustMF [19]: represents the influence of connections to target user preferences in two ways: truster and trustee. This approach provides recommendations to users that usually exhibit influence on others and those who are typically influenced by others.

D. Parameter Settings

We use LibRec [20], which provides an implementation of SoRec, SocialMF and TrustMF with default parameters. The implementation of ToSocialRec was done on top of BiasedMF algorithm also in LibRec library. Therefore, we cluster BiasedMF prediction models and include Preference-like score in the recommendation process.

Experiments were executed with 10 latent factors and 100 iterations for the model building phase. The social information weight are measured by λ_c, β and λ_t in SoRec, SocialMF and TrustMF respectively, we were varying them between 0.1 and 100. Optimal experimental settings for λ_c is equal to 20 in Facebook, 50 in Flixster, 100 in Filmtrust and 1 in Epinions, Paintings and Clothing. SocialMF achieves better results with β equal to 100 in Facebook and FilmTrust, 1 in Flixster, 50 in Epinions, 0.5 in Paintings and 0.1 in Clothing. Finally, λ_t has optimal values equal to 100 in Facebook, Filmtrust, Epinions and Paintings, 0.9 in Flixster and 20 in Clothing.

With ToSocialRec, we also experimentally test several cluster sizes. Then we set the optimal number of clusters to 6 clusters for FilmTrust, 7 clusters for Epinions, 4 clusters for Facebook, 9 clusters for Flixster, 3 clusters for Paintings and 5 clusters for Clothing dataset. Beside this, we apply K-means (using the Euclidean distance measure) as the clustering algorithm. Minimum threshold ϵ has optimal values for Similarity equal to 0.4 and 0.2 in Facebook and Flixster. VP-Sim. achieves better results with $\epsilon = 0.7$ for Paintings and $\epsilon = 0.5$ to Clothing. Centrality has optimal ϵ value equal to 0.1 for Facebook, Flixster, FilmTrust and Epinions. While Mutual has best results to $\epsilon = 0.4$ for Facebook, 0.2 for Flixster and Epinions and 0.1 to FilmTrust.

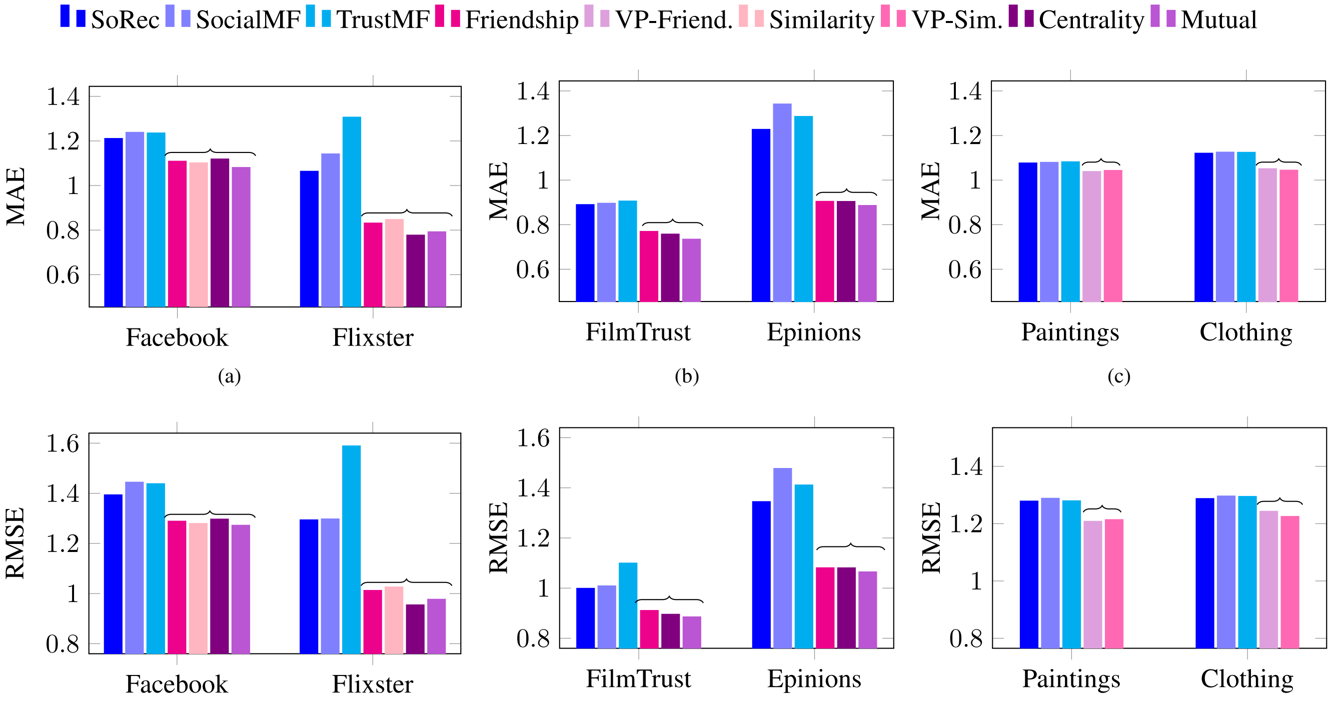
E. Results and Discussion

Figure 3 presents the histograms of MAE (upper part) and RMSE (lower part) for each approach, and for all datasets. Recall that those metrics are negatively-oriented scores: lower values are better.

We can see that ToSocialRec methods perform better than the other methods it is compared to: SoRec, SocialMF, and TrustMF. Specifically, comparing the best result among of these three state of the art algorithms against ours, we note the following improvements in MAE per each dataset: 10.78% on Facebook, 27.22% on Flixster, 17.61% on FilmTrust, 26.43% on Epinions, 2.74% on Paintings, and 6.72% on Clothing.

Results in terms of RMSE were similar to MAE. The improvements in RMSE are: 8.73% on Facebook, 22.29% on Flixster, 11.42% on FilmTrust, 19.69% on Epinions, 4.07% on Paintings, and 4.85% on Clothing.

Table V presents the $nDCG$ at rank positions 5, 10, 15, and 20. For each approach and for each dataset, the largest values is indicated by boldface in each column . The lower part of each table shows the results of ToSocialRec using different Preference-like score methods. Although we can see that ToSocialRec is performing better than the other methods, the difference is quite small on some datasets. For instance, Table VIa presents SoRec achieving 0.8537 for $nDCG@5$ while ToSocialRec Centrality score is 0.8541.



(a) (b) (c)

(d) (e) (f)

Fig. 3. MAE and RMSE histograms for each approach per dataset under cold-start scenario (0-rating protocol). Please note that in each pane, the 3 leftmost blueish bars are algorithms we compare ToSocialRec to, while the pinkisk rightmost bars (over-braced) are variant of ToSocialRec.

Statistical Analyses. We checked the normality and homogeneity of the results for each method for each metric (MAE, RMSE, and nDCG) using Shapiro and Bartlett test. We observed that the results are not normally distributed and not homogeneous. Therefore, we performed the global comparisons with Kruskal-Wallis test.

As measured in terms of MAE and RMSE, ToSocialRec produces better results with 95% of confidence. Remarkably, between *Mutual and Centrality* methods (our best results), there is no statistically significant difference. The analysis of nDCG brings slightly different results. Although ToSocialRec, using *Mutual, Centrality, and Friendship*, produced overall significantly better nDCG results, *SoRec* scores the same as *VP-Friendship* and *VP-Similarity*, again with a 95% confidence level.

We asked whether a recommender system, based on its current set of prediction models that have been built considering current users, might offer accurate recommendations for new/cold users. To this question, it seems the answer is positive.

RQ#1: How well can a selected model predicts the ratings of new users?

The use of existing prediction models pays off: by exploiting them, ToSocialRec can predict item rating better than a personalized new one built with just social information.

RQ#2: How well can a selected model rank items for a new user?

Two of our Preference-like functions did not present better results. However, selecting a model instead of build a new one can still lead to high quality items ranking.

Our experimental results on real datasets indicates that ToSocialRec performs better or at least equivalently as all methods we compare to, particularly, for the cold-start user.

Limitations. The insights from this work are limited by the methodology and the dataset that have been used. The main limitation is that we conducted our experiments comparing Matrix Factorization approaches. While we test on real, diverse and well studied datasets, ToSocialRec might not yet perform better than other recommender methods. Future work could compare a larger set of state of the art systems.

Another threat arises from the randomized sampling of the original Flixter and Epinions datasets. Due to computational resource limitations we deliberately reduce the size of those datasets. The problem is about parameter selection. For example, setting the number of clusters requires running the model with multiple parameter values and then selecting the best one. However, we feel rather confident about our results because we test our hypothesis on four other datasets. In addition, ToSocialRec achieves better results on sparser datasets, which is the situation met in real applications. Future work could investigate not only how to set the parameters in a more scalable way, but might evaluate the approach against very large datasets.

TABLE V
nDCG FOR COLD-START SCENARIO (0-RATING PROTOCOL).

Approach	Rank size			
	@5	@10	@15	@20
SoRec	0.8537 ± .132	0.8457 ± .117	0.8441 ± .107	0.8424 ± .109
SocialMF	0.8226 ± .136	0.8205 ± .122	0.8202 ± .114	0.8240 ± .115
TrustMF	0.8509 ± .136	0.8445 ± .118	0.8427 ± .109	0.8428 ± .109
Friendship	0.8549 ± .130	0.8475 ± .114	0.8451 ± .106	0.8437 ± .107
Similarity	0.8562 ± .131	0.8490 ± .114	0.8469 ± .106	0.8454 ± .107
Centrality	0.8541 ± .130	0.8447 ± .115	0.8433 ± .106	0.8461 ± .108
Mutual	**0.9054** ± .076	**0.8869** ± .064	**0.8840** ± .055	**0.8716** ± .055

(a) Facebook

Approach	Rank size			
	@5	@10	@15	@20
SoRec	0.8226 ± .127	0.8197 ± .115	0.8171 ± .110	0.8156 ± .107
SocialMF	0.7416 ± .138	0.7415 ± .125	0.7437 ± .119	0.7461 ± .115
TrustMF	0.7204 ± .135	0.7246 ± .122	0.7270 ± .117	0.7298 ± .113
Friendship	0.8344 ± .125	0.8291 ± .113	0.8259 ± .108	0.8242 ± .105
Similarity	0.8376 ± .122	0.8331± .112	0.8306 ± .107	0.8292 ± .104
Centrality	0.8388 ± .117	0.8343 ± .106	0.8315 ± .102	0.8298 ± .099
Mutual	**0.8530** ± .094	**0.8428** ± .093	**0.8382** ± .088	**0.8384** ± .087

(b) Flixster

Approach	Rank size			
	@5	@10	@15	@20
SoRec	0.8432 ± .129	0.8447 ± .111	0.8458 ± .103	0.8476 ± .097
SocialMF	0.8444 ± .129	0.8460 ± .110	0.8476 ± .102	0.8489 ± .096
TrustMF	0.8146 ± .129	0.8203 ± .111	0.8277 ± .101	0.8346 ± .092
Friendship	0.8456 ± .124	0.8500 ± .110	0.8517 ± .100	0.8547 ± .094
Centrality	0.8489 ± .117	0.8527 ± .102	0.8549 ± .094	0.8605 ± .089
Mutual	**0.8633** ± .108	**0.8586** ± .096	**0.8605** ± .089	**0.8614** ± .084

(c) FilmTrust

Approach	Rank size			
	@5	@10	@15	@20
SoRec	0.9093 ± .089	0.9108 ± .069	0.9141 ± .059	0.9181 ± .054
SocialMF	0.8983 ± .095	0.9021 ± .073	0.9063 ± .062	0.9109 ± .057
TrustMF	0.8222 ± .130	0.8217 ± .109	0.8262 ± .097	0.8332 ± .089
Friendship	0.9141 ± .085	0.9144 ± .065	0.9172 ± .055	0.9224 ± .050
Centrality	0.9146 ± .085	0.9147 ± .065	0.9173 ± .055	0.9226 ± .050
Mutual	**0.9189** ± .084	**0.9170** ± .068	**0.9196** ± .056	**0.9236** ± .052

(d) Epinions

Approach	Rank size			
	@5	@10	@15	@20
SoRec	0.8332 ± .126	0.8301 ± .110	0.8258 ± .101	0.8219 ± .098
SocialMF	0.7187 ± .153	0.6961 ± .130	0.6818 ± .117	0.6663 ± .113
TrustMF	0.6524 ± .145	0.6576 ± .130	0.6668 ± .121	0.6736 ± .115
VP-Friend.	0.8403 ± .124	0.8307 ± .112	0.8289 ± .101	0.8232 ± .098
VP-Sim.	**0.8434** ± .127	**0.8329** ± .113	**0.8313** ± .101	**0.8257** ± .098

(e) Paintings

Approach	Rank size			
	@5	@10	@15	@20
SoRec	0.7662 ± .157	0.7559 ± .137	0.7572 ± .128	0.7632 ± .119
SocialMF	0.7715 ± .153	0.7638 ± .134	0.7628 ± .125	0.7715 ± .120
TrustMF	0.7684 ± .147	0.7676 ± .129	0.7677 ± .123	0.7726 ± .118
VP-Friend.	0.7769 ± .152	0.7703 ± .130	0.7731 ± .122	0.7744 ± .113
VP-Sim.	**0.7785** ± .151	**0.7709** ± .134	**0.7732** ± .124	**0.7757** ± .116

(f) Clothing

V. Related Work

This paper proposes an approach to mitigate user cold-start problem selecting prediction models within a recommendation system. The selection is done through social and non-social user network. Thus, this work concerns two research fields:

User cold-start problem. Recommending appropriate items to a cold user is well-known to be challenging and research on this topic spans more than a decade [21]. Recently, authors have turned themselves towards designing hybrid recommender systems. Son proposed a hybrid method, called HU-FCF++, that combines features of existing methods based on clustering techniques and similarity metrics [22]. Pereira and Hruschka envisioned a hybrid recommendation method to address the cold-start problem based on the simultaneous co-clustering and learning of user and item attributes [3]. Lika et al. also proposed a user cold-start recommendation method [23]. Their idea is to classify a cold-start user in a set of categories based on his demographic information. Thus, according to demography similarity and neighbors profile, the system provides recommendations.

In contrast with these works, we focus on selecting a consensual prediction model, built with a matrix factorization technique and applying cluster algorithms. We assume the existence of a preference-like network with the preference like score between users as a way to select the most suitable model for a cold-start user.

Social Recommender Systems. An interesting source of information to recommender system is social data, mainly, because harvesting social information on the web has become very common and effective [24]. Hao Ma et al. [18], [25] have proposed social information enhanced algorithms to improve matrix factorization based recommenders. For instance, SoRec [18] relies on probabilistic matrix factorization, to better deal with data sparsity and accuracy problems. As SoRec is reported by its authors to achieve high accuracy recommendations for cold users, we include it in our experimental study.

Besides SoRec, as described in Section IV-C, we also compare our results against two other approaches: TrustMF [19], which is an adaptation of matrix factorization technique to map users in terms of their trust relationship; and SocialMF [13], which explores the propagation of trust among users. Both systems present high scores dealing with cold-start users and that is why we compared them against our approach.

Furthermore, in terms of assuming that users' social network structure reflects actual similarities among users, there are common aspects between ToSocialRec and the work of Delporte et al. [6]. They developed an improved matrix factorization based recommender combining social information with implicit feedback. In comparison, our goal is to explore the numerical value of explicit feedback, which indicates preferences. Besides, we do not use social information to build the prediction model.

In spite of many years of research, research on social recommendation systems is still very active. Recent works are reporting new findings exploring data sparsity [1], social relationship [26], and new techniques, such as genetic algorithms [2]. We previously proposed SOCIAL PREFREC [9], [10] that aims at exploiting social networks in pairwise preference fashion. Here, we expanded that work by generalizing and introducing the concept of network similarity to model based recommender systems.

VI. CONCLUSION

Social information is often massive, and social recommender system (SRS) are already taking advantage of this source of information. In this work, we proposed a novel approach to exploit existing prediction models instead of creating new ones, which allows improving the recommendations for cold-start users. We studied the network metrics in social and non-social contexts. We found that using a similarity score, dubbed Preference-like, among users of a recommender system is capable to accurately recommend items for new users.

The dominant trend in SRS has been towards designing new prediction models using social data. However, in many real-life situations, integrating new models into legacy systems may not be possible. Furthermore, the results of this paper suggest that it can be fruitful to explore the predictions and users already using the recommendation system. The experiments provided statistical evidences that these existing models and users hold enough information to lead to more accurate item recommendations for cold users.

Overall, this paper makes the following main contributions:

1) We compare and contrast several matrix factorization based social recommender systems in a cold user scenario;
2) We propose a general model selection approach that leverages existing prediction models to offer items for new users;
3) We thoroughly evaluate our approach on six distinct datasets and show its effectiveness in contrast to state-of-the-art matrix factorization recommenders.

ACKNOWLEDGMENTS

C. Z. Felício thanks the Federal Institute of Triângulo Mineiro for study leave granted. We also thank the Brazilian Research Agencies CAPES, CNPq and FAPEMIG. Ph. Preux 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 support of Inria, and the exciting intellectual environment provided by SequeL.

REFERENCES

[1] I. Barjasteh, R. Forsati, D. Ross, A. H. Esfahanian, and H. Radha, "Cold-start recommendation with provable guarantees: A decoupled approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1462–1474, June 2016.

[2] D. H. Alahmadi and X. J. Zeng, "Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis," in *IEEE 27th International Conf. on Tools with Artificial Intelligence (ICTAI)*, Nov 2015, pp. 1045–1052.

[3] A. L. V. Pereira and E. R. Hruschka, "Simultaneous co-clustering and learning to address the cold start problem in recommender systems," *Knowledge-Based Systems*, vol. 82, pp. 11–19, 2015.

[4] A. Q. Macedo, L. B. Marinho, and R. L. Santos, "Context-aware event recommendation in event-based social networks," in *Proc. 9th ACM Conf. on Recommender Systems*, ser. RecSys '15, 2015, pp. 123–130.

[5] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo, "A reusable methodology for the instantiation of social recommender systems," in *IEEE 25th International Conf. on Tools with Artificial Intelligence (ICTAI)*, Nov 2013, pp. 775–782.

[6] J. Delporte, A. Karatzoglou, T. Matuszczyk, and S. Canu, *Machine Learning and Knowledge Discovery in Databases: European Conf. ECML PKDD, Prague, Czech Republic, September, Proc.* Berlin, Heidelberg: Springer, 2013, ch. Socially Enabled Preference Learning from Implicit Feedback Data, pp. 145–160.

[7] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proc. 7th ACM SIGCOMM Conf. on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 29–42.

[8] R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*. New York, NY, USA: Cambridge University Press, 2014.

[9] C. Z. Felício, K. V. R. Paixão, G. Alves, and S. de Amo, "Social prefrec framework: leveraging recommender systems based on social information," in *Proc. 3rd Symposium on Knowledge Discovery, Mining and Learning*, 2015, pp. 66–73.

[10] C. Z. Felício, K. V. R. Paixão, G. Alves, S. de Amo, and P. Preux, "Exploiting social information in pairwise preference recommender system," *Journal of Information and Data Management*, (To appear).

[11] C. Z. Felício, C. M. M. de Almeida, G. Alves, F. S. F. Pereira, K. V. R. Paixão, and S. de Amo, *Advances in AI: 29th Canadian Conf. on Artificial Intelligence, Victoria, BC, Canada, May.* Springer, 2016, ch. Visual Perception Similarities to Improve the Quality of User Cold Start Recommendations, pp. 96–101.

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[13] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. Fourth ACM Conf. on Recommender Systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 135–142.

[14] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," in *Proc. 23rd International Joint Conf. on Artificial Intelligence (IJCAI)*, 2013, pp. 2619–2625.

[15] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. 2007 ACM Conf. on Recommender Systems*, ser. RecSys '07. New York, NY, USA: ACM, 2007, pp. 17–24.

[16] E. Viriato de Melo, E. A. Nogueira, and D. Guliato, "Content-based filtering enhanced by human visual attention applied to clothing recommendation," in *Tools with Artificial Intelligence (ICTAI), IEEE 27th International Conf. on*, Nov 2015, pp. 644–651.

[17] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010, ch. Leave-One-Out Cross-Validation, pp. 600–601.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. on Information and Knowledge Management*, ser. CIKM '08. New York, NY, USA: ACM, 2008, pp. 931–940.

[19] B. Yang, Y. Lei, D. Liu, and J. Liu, "Social collaborative filtering by trust," in *Proc. Twenty-Third International Joint Conf. on Artificial Intelligence*, ser. IJCAI. AAAI Press, 2013, pp. 2747–2753.

[20] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A java library for recommender systems," in *23rd Conf. on User Modeling, Adaptation, and Personalization (UMAP)*, 2015.

[21] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 253–260.

[22] L. H. Son, "Hu-fcf++: A novel hybrid method for the new user cold-start problem in recommender systems," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 207 – 222, 2015.

[23] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[24] I. Guy, *Recommender Systems Handbook*. Boston, MA: Springer US, 2015, ch. Social Recommender Systems, pp. 511–543.

[25] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving recommender systems by incorporating social contextual information," *ACM Trans. Inf. Syst.*, vol. 29, no. 2, pp. 9:1–9:23, Apr. 2011.

[26] W. Reafee, N. Salim, and A. Khan, "The power of implicit social relation in rating prediction of social recommender systems," *PLoS ONE*, vol. 11, no. 5, pp. 1–20, 05 2016.