Computer Communications 41 (2014) 1–10

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A survey of collaborative filtering based social recommender systems

Xiwang Yang[a,*], Yang Guo[b], Yong Liu[a], Harald Steck[c,1]

[a] Polytechnic Institute of NYU, Brooklyn, NY, USA
[b] Bell Labs, Alcatel-Lucent, Holmdel, NJ, USA
[c] Netflix Inc., Los Gatos, CA, USA

ARTICLE INFO

Article history:
Received 22 July 2012
Received in revised form 4 June 2013
Accepted 26 June 2013
Available online 16 July 2013

Keywords:
Social network
Recommender system
Collaborative filtering

ABSTRACT

Recommendation plays an increasingly important role in our daily lives. Recommender systems automatically suggest to a user items that might be of interest to her. Recent studies demonstrate that information from social networks can be exploited to improve accuracy of recommendations. In this paper, we present a survey of collaborative filtering (CF) based social recommender systems. We provide a brief overview over the task of recommender systems and traditional approaches that do not use social network information. We then present how social network information can be adopted by recommender systems as additional input for improved accuracy. We classify CF-based social recommender systems into two categories: matrix factorization based social recommendation approaches and neighborhood based social recommendation approaches. For each category, we survey and compare several representative algorithms.

1. Introduction

Communication networks facilitate easy access of information. Meanwhile, the richness of online information also brings forth the "information overload" problem. For example, if one wants to buy a digital camera, it would be a frustrating experience for her to read through and compare all online reviews about digital cameras before making the purchase decision. Recommender systems deal with information overload by automatically suggesting to users items that may fit their interests. Accurate recommendations enable users to quickly locate desirable items without being overwhelmed by irrelevant information. It is also of great interest for vendors to recommend those products that match the interests of each of the visitors of their websites, and hopefully turn them into satisfied and returning customers. No wonder, in the Netflix contest [19], an improvement of 10% recommendation accuracy was awarded with 1 million US dollars.

Recommender system (RS) roots back to several related research disciplines, such as cognitive science, approximation theory and information retrieval, etc. Due to the increasing importance of recommendation, it has become an independent research field since the mid 1990s [1]. Broadly speaking, a RS suggests to a user those items that might be of her interest. Generally, there are two variants of recommendation approaches: content-based and collaborative-filtering (CF) based approaches [1,2]. CF approaches can be further grouped into model-based CF and neighborhood-based CF [2,3]. Model-based approaches use user-item ratings to learn a predictive model. The general idea is to model the user-item interactions with factors representing latent features of users and items in the system, such as the preference class of users and the category class of items. In contrast, neighborhood-based CF approaches use user-item ratings stored in the system to directly predict ratings for new items.

Online social networks (OSN) present new opportunities as to further improve the accuracy of RSs. In real life, people often resort to friends in their social networks for advice before purchasing a product or consuming a service. Findings in the fields of sociology and psychology indicate that humans tend to associate and bond with similar others, also known as *homophily* [25]. Due to stable and long-lasting social bindings, people are more willing to share their personal opinions with their friends, and typically trust recommendations from their friends more than those from strangers and vendors. Popular online social networks, such as Facebook [21], Twitter [20], and Youtube [17], provide novel ways for people to communicate and build virtual communities. Online social networks not only make it easier for users to share their opinions with each other, but also serve as a platform for developing new RS algorithms to automate the otherwise manual and anecdotal social recommendations in real-life social networks.

A social RS improves on the accuracy of the traditional RS by taking social interests and social trusts between users in an OSN as additional inputs. For example, due to social interest, a user may read a particular news article about an event just because

[*] Corresponding author.
E-mail addresses: xy271@nyu.edu (X. Yang), Yang.Guo@alcatel-lucent.com (Y. Guo), yongliu@poly.edu (Y. Liu), hsteck@netflix.com (H. Steck).
[1] Part of this work was done while at Bell Labs, Alcatel-Lucent, Murray Hill, NJ.

0140-3664/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.06.009

the event occurred in a place where her family lives; due to social trust, a user may like a song recommended by her close friends on Facebook. Social trust between a pair of friends (u, v) may be established based on explicit feedback of user u concerning user v (e.g., by voting), or it may be inferred from implicit feedback (e.g., the frequency and quantity of interaction/communication/email exchanges between u and v). Different social RS algorithms explore social networks and the embedded social information differently.

In this survey, we focus on CF-based social RSs, since most existing social recommender systems are CF-based. Following the classification of traditional CF-based RSs [2,3], we classify CF-based social RSs into two main categories: *Matrix Factorization (MF) based social recommendation approaches* and *Neighborhood based social recommendation approaches.* In MF-based social recommendation approaches, user-user social trust information is integrated with user-item feedback history (e.g., ratings, clicks, purchases) as to improve the accuracy of traditional MF-based RSs, which only factorize user-item feedback data. *Neighborhood based social recommendation approaches* include *Social Network Traversal (SNT) based approaches* and *Nearest Neighbor (NN) approaches*. A SNT-based algorithm synthesizes a recommendation for a user after traversing and querying her direct and indirect friends in her neighborhood in the social network. A *NN* approach combines the traditional CF neighborhood with social neighborhood, and predicts ratings of items or recommends a list of items.

The rest of this survey is organized as follows. We formally present the task of RS in Section 2. The traditional CF-based RS is briefly presented in Section 3. We then introduce online social network as an additional RS input in Section 4. MF-based social recommendation approaches are surveyed in Section 5. Neighborhood based social recommendation approaches are surveyed in Section 6. We make a comparison of CF-based social recommendation approaches in Section 7. The survey is summarized in Section 8.

2. Task of recommender systems

Recommender systems typically provide a user with a list of recommended items she may be interested in, or predict how much she might prefer each item. These systems help users to decide on appropriate items, and ease the task of finding preferred items in a collection.

The main body of literature has been concerned with the accuracy of predicting rating values. To this end, test data is represented as a user-item rating-matrix $R \in \mathbb{R}^{u_0 \times i_0}$, where u_0 denotes the number of users, and i_0 the number of items. $R_{u,i}$ is user *u's* rating value of item *i*. Typically, there are lots of missing values in the user-item rating-matrix R. The sparsity of R often is larger than 99% in commercial systems [42]. Table 1 illustrates a toy rating-matrix concerning six users (denoted as u_1 to u_6) and seven items (denoted as i_1 to i_7). Each user rates some items as to express her interests in each of the items. The ratings are often on a numerical five-star scale, where one and two stars represent negative ratings, three stars represent ambivalence, while four and five stars represent positive ratings. A RS algorithm predicts the missing ratings in the matrix, and recommends an item to a user if her predicted rating for the item is, say, four or five stars.

Table 1
User-Item Rating-Matrix.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	5		1	5			2
u_2	4	1		5		4	1
u_3	5		1		5	5	1
u_4			5			3	
u_5	2						5
u_6		2				5	

Typically, the rating data-set is split into a training set and a test set, where the training set is used for model fitting and parameter tuning, and the test set serves for evaluating the RS. Let the matrix of the predicted ratings be denoted as $\widehat{R} \in \mathbb{R}^{u_0 \times i_0}$. As to assess the accuracy of an RS, the most popular evaluation metrics are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE):

$$RMSE = \sqrt{\frac{\sum\limits_{(u,i)\in\mathcal{R}_{test}} \left(R_{u,i} - \widehat{R}_{u,i}\right)^2}{|\mathcal{R}_{test}|}}, \tag{1}$$

$$MAE = \frac{\sum_{(u,i)\in\mathcal{R}_{test}} |R_{u,i} - \widehat{R}_{u,i}|}{|\mathcal{R}_{test}|}, \tag{2}$$

where $\mathcal{R}_{test}$ is the set of all user-item pairs (u, i) in the test set. The lower the RMSE/MAE is, the closer the predicted ratings are to the actual ratings on average.

Instead of presenting predicted item-ratings to users, commercial RS algorithms normally provide a user with a list of k recommended items she might prefer, also known as *Top-k Recommendation*. Instead of RMSE and MAE, the direct accuracy measures of top-k RSs are *top-k hit ratio (or recall)*, *precision* and *Normalized Discounted Cumulative Gain (NDCG)* [43], among others. As to compute the top-k hit-ratio or recall, for each user u, we rank the items according to the predicted rating $\widehat{R}_{u,i}$ or a voting value. Here we use predicted rating as an example. If the predicted rating is continuous, the ranking list is unique. Otherwise, ties may be broken at random. An item is defined as *relevant* to a user in the test set if she finds it appealing or interesting (e.g., the assigned rating in the test data is above a certain threshold). For instance, the rating values range from 1 to 5 stars in the Netflix [19] data, and the author in [9] considered 5-star ratings as relevant (i.e. the user definitely liked these items), while other rating values and missing rating values are considered irrelevant. Other choices led to similar results. Now the top-k hit ratio or recall can be defined as the fraction of relevant items in the test set that is in the top-k of the ranking list, denoted by $N(k, u)$, from among all relevant items in the test set for user u, denoted by $N(u)$. For each user u, the top-k hit ratio is given by

$$\mathrm{H}(k, u) = \frac{N(k, u)}{N(u)}. \tag{3}$$

It can be aggregated over all users as to obtain the average top-k hit ratio or recall for the test set, for instance

$$recall(k) = \frac{\sum_u N(k, u)}{\sum_u N(u)}, \tag{4}$$

which is a weighted average across users, with each user's weight proportional to the user's number of relevant items $N(u)$. A higher top-k hit ratio or recall suggests more accurate top-k recommendations. Top-k hit ratio and recall are non-decreasing functions of k. Precision is another popular evaluation metric for top-k recommendations. For each user, precision is given by

$$precision(k, u) = \frac{N(k, u)}{k}, \tag{5}$$

which can be interpreted as the fraction of relevant items among the k items recommended to user u. For a given user and fixed k, precision is proportional to recall. We aggregate precision over all users to obtain the average precision for the test set as follows:

$$precision = \frac{1}{u_0}\sum_u \frac{N(k,u)}{k} = \frac{\sum_u N(k,u)}{u_0 k}, \tag{6}$$

where u_0 is the number of users. NDCG is another accuracy measure from information retrieval, where the gain of a recommended item is discounted logarithmically with respect to its position/ranking in the whole recommendation list [43]. Specifically, assuming that each user u has a gain of $g_{u,i}$ when item i is recommended, the Discounted Cumulative Gain (DCG) of a list of k items for user u is defined as:

$$DCG@k(u) = \sum_{j=1}^{k} \frac{g_{u,i(j)}}{max(1, log_b j)}, \tag{7}$$

where $i(j)$ denotes the j-th item in the ordered list of recommendations, and the logarithmic base b is a free parameter, typically between 2 and 10. A logarithm with base 2 is commonly used to ensure all positions are discounted.

NDCG of user u is the normalized version of DCG, given by:

$$NDCG@k(u) = \frac{DCG@k(u)}{DCG^*@k(u)}, \tag{8}$$

where $DCG^*@k(u)$ is the ideal DCG@k (u), i.e., items are sorted in decreasing order with respect to real $R_{u,i}$, and the list is truncated at position k.

The average DCG of a list of k items is defined as:

$$DCG@k = \frac{1}{u_0}\sum_{u=1}^{u_0} DCG@k(u).$$

Similarly, the average NDCG of a list of k items is defined as:

$$NDCG@k = \frac{1}{u_0}\sum_{u=1}^{u_0} NDCG@k(u).$$

Currently, most RS algorithms have been evaluated and ranked on their prediction powers, i.e., their capabilities to accurately predict the user's choices. However, it is now widely agreed upon that prediction accuracy is crucial, but insufficient by itself for good real-world RSs [57,58]. In many applications, people use a RS for more than an exact anticipation of their tastes. Users may also be interested in discovering new and diverse items, deviating from their routine choices. It is also important for a RS to preserve users' privacy when coming up with good recommendations. The responsiveness of a RS is critical if items to be recommended are highly dynamic, such as news articles. It is hence important to identify the set of relevant properties that may influence the success of a RS in a specific application context.

3. Recommender-systems based on feedback-data

Social network information has become available only recently as to improve recommender systems. Before we outline the various approaches of using social network information, we briefly review two main variants in the following: content-based approaches and collaborative-filtering approaches.

The basic idea of the content-based approach is to use properties of an item to predict a user's interest towards it. For instance, for a book, one may use the author's name, the genre, keywords and tags. These properties are then matched to the taste of a target user.

The key idea of collaborative filtering is to use the feedback from each individual user. Concerning the user's feedback, one may distinguish between explicit feedback (e.g., the user assigns a rating to an item) and implicit feedback (e.g., the user clicks on a link, listens to a song, or purchases an item). When data concerning a sufficiently large number of users and their feedback is available, it can be used to determine similar users (e.g., users who listened to the same collection of songs); items can then be recommended among similar users: while similar users have a large overlap in their collections of songs, each user may have listened to a few additional songs; those additional songs can then be recommended to the other similar users. Complementary to this idea of identifying similar users based on the similarity of their past behaviors, similarity between items can be inferred analogously, i.e., when they are, say, purchased by the same user (s). This basic idea underlying collaborative filtering was found to lead to very accurate recommendations in the literature [6,7,44,47]. Since most of the existing social recommender systems are CF-based, we focus on CF-based traditional RSs in this section.

Much work in the literature has focused on using explicit feedback data, in particular ratings (e.g. on a scale from one to five stars) assigned to items (e.g., movies, songs, restaurants). The task was to predict the rating values of other items rated by a user, and recommendation accuracy was typically measured in terms of RMSE or MAE. Following the basic idea underlying the collaborative filtering approach, various kinds of nearest neighbor approaches have been proposed, which can be categorized into user-user and item-item neighborhood models and combinations thereof, e.g., see [6]. One of the most accurate approaches was found to be matrix factorization [24,5,6,13]. This approach had been found useful in computer vision [16] and text analysis [15] before. The most basic approach to matrix factorization is singular value decomposition, but numerous more sophisticated approaches have been developed, e.g., [24,5,6,13]. The underlying idea is to map the users and items into a low-dimensional latent space, and to determine similarities between users and items in this latent space. For instance, the matrix of predicted ratings $\widehat{R} \in \mathbb{R}^{u_0 \times i_0}$may be modeled as follows

$$\widehat{R} = r_m + QP^\top, \tag{9}$$

with matrices $P \in \mathbb{R}^{i_0 \times j_0}$ and $Q \in \mathbb{R}^{u_0 \times j_0}$, where $j_0 \ll i_0, u_0$ determines the (low) rank (e.g., 50); and $r_m \in \mathbb{R}$ is a (global) offset. $Q \in \mathbb{R}^{u_0 \times j_0}$ essentially represents the latent user profiles, while $P \in \mathbb{R}^{i_0 \times j_0}$ captures the latent item profiles. Using gradient descent methods, they can be determined, for instance, by minimizing the squared error between the given rating value $R_{u,i}$ and the value $\widehat{R}_{u,i}$ predicted by the model for users u and items i:

$$\sum_u \sum_i W_{u,i} \cdot \left(R_{u,i} - \widehat{R}_{u,i}\right)^2 + \lambda\left(||P||_F^2 + ||Q||_F^2\right), \tag{10}$$

where the last term is added to regularize the learned matrices P and Q as to prevent over-fitting; $\lambda > 0$ is the regularization parameter, and the Frobenius norm is denoted by $||\cdot||_F$. There are various ways of specifying the training weights $W_{u,i}$. A simple but effective choice is

$$W_{u,i} = \begin{cases} 1 & \text{if } R_{u,i} \text{ observed}, \\ w_m & \text{otherwise}. \end{cases} \tag{11}$$

When the objective is to optimize RMSE on the observed ratings, then $w_m = 0$. If the objective is to obtain a good ranking of all items (as measured, e.g., by precision, recall or NDCG), then a small value $w_m > 0$ is advantageous [9,53], combined with imputing a low value for unobserved $R_{u,i}$.

The matrix factorization approach can also be depicted as a probabilistic graphical model [8], as illustrated in Fig. 1. The rating value $R_{u,i}$ is obtained by combining the matrices Q and P. The prior distributions for the entries in the matrices Q and P are denoted by σ_Q and σ_P; this results in the L2 regularization term in the equation above, see [8] for details. The prior over the rating values, depicted by σ_R, gives rise to the weight w_m.

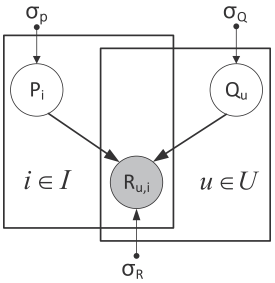

Fig. 1. The BaseMF graphical model.

Table 2
User-User Trust Matrix.

	u_1	u_2	u_3	u_4	u_5	u_6
u_1		0.9	0.7			
u_2	0.8					0.7
u_3		0.8				
u_4			0.2		0.9	
u_5				0.8		
u_6	0.6				0.4	

Matrix factorization has also been combined with the neighborhood approach [6]. The conditional restricted Boltzmann machine [14] is yet another very successful model. Recommendation accuracy can be further improved by using an ensemble of different models, whose predictions get combined in a final blending step. Various approaches for blending have been developed, e.g. see [4].

Using implicit feedback has received considerably less attention in the literature. Notable publications in this domain include [10,12,11], which aim at recommending TV shows to users based on their past viewing behaviors, such as how much time they spent on each type of TV programs. As implicit feedback like this is typically much more abundant than explicit feedback data in practical applications, RSs based on implicit feedback are typically deployed.

4. Social network as additional RS input

Now we survey how information from social networks can be adopted by RS algorithms. We assume that users are connected in an underlying social network, either a general-purpose social network, such as Facebook [21], or a domain-specific recommendation social network, such as Flixster [23] for movie recommendations and Epinions [18] for a wide range of product recommendations. We denote the underlying social network as a directed graph $G = (U, F)$, where U is the set of users with $|U| = u_0$, and F is the set of friendship links. The social network information is represented by a matrix $S \in \mathbb{R}^{u_0 \times u_0}$. Each user u has a set $\mathcal{F}_u^+$ of direct neighbors that u trusts or follows, and at the same time, u is trusted/followed by a set $\mathcal{F}_u^-$ of users. The directed and weighted social relationship of user u with user v (e.g. user u trusts/knows/follows user v) is represented by a positive value $S_{u,v} \in (0, 1]$. An absent or unobserved social relationship is reflected by $S_{u,v} = s_m$, where typically $s_m = 0$. The social weight $S_{u,v}$ can be interpreted as how much user u trusts or knows user v in a social network. It may be based on explicit feedback of user u concerning user v (e.g., by voting), or inferred from implicit feedback (e.g., the degree of interaction/communication). Normally, social trust $S_{u,v}$ is non-negative. In special cases, it can also take negative values, explicitly modeling the two users' conflicting tastes. Fig. 2 illustrates a toy-example of a social network between six users, each of which has a set of friends. Each directed friendship-link is weighted by a positive trust value. The social trusts between all user-pairs are captured by the matrix S illustrated in Table 2. In this paper, trust network and social network are used interchangeably as general terms.

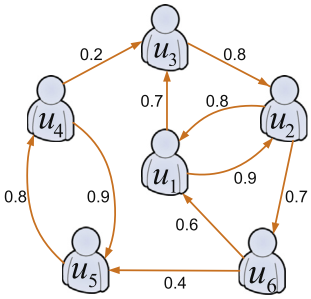

Fig. 2. Social Network Graph.

4.1. Social circles for RS

Most existing social RSs mine a social network as a whole. Recently, authors of [52] proposed circle-based recommendations in online social networks. It is obvious that a user's social life, being online or offline, is intrinsically multifaceted. Intuitively, a user trusts different subsets of friends in different domains. For example, in the context of multi-category recommendation, a user u may trust user v in the Cars category while not trusting v in the Kids' TV Show category. Therefore, u should care less about v's ratings in the Kids' TV Show category than in the Cars category. Ideally, if we know a user's trust circles in different categories, we probably should only use her trust circles specific to the category for which we want to predict ratings. Unfortunately, in most existing multi-category rating data-sets, a user's social connections are mixed together from all categories. Even if the circles were explicitly known, e.g. Circles in Google+ [22] or Facebook [21], they may not correspond to particular item categories that a recommender system may be concerned with.

In [52], the authors proposed a set of algorithms to infer category-specific circles of friends, and to infer the trust value on each link based on users' rating activities in each category. They infer the circles of friends from rating (or other feedback) data concerning items *that can be divided into different categories (or genres etc.)*. The basic idea is that a user may trust each friend only concerning certain item categories but not regarding others. They divide a social network S of all trust relationships into several sub-networks $S^{(c)}$, each of which concerning a single category c of items.

Definition – Inferred Circle: *Regarding each category c, a user v is in the inferred circle of user u, i.e., in the set* $\mathcal{C}_u^{(c)}$, *if and only if the following two conditions hold:*

- $S_{u,v} > 0$ *in the (original) social network, and*
- $N_u^{(c)} > 0$ *and* $N_v^{(c)} > 0$ *in the rating data,*

where $N_u^{(c)}$ *denotes the number of ratings that user u has assigned to items in category c. Otherwise, user v is not in the circle of u concerning category c, i.e.,* $v \notin \mathcal{C}_u^{(c)}$.

This is illustrated in a toy example in Fig. 3. They further proposed a set of algorithms to construct the trust value $S_{u,v}^{(c)}$ of user u to each friend v in her trust circle $\mathcal{C}_u^{(c)}$. When it comes to predict users' rating of an item in one category, only a circle that corresponds to this item's category is used as social network input. Recommendation methods that use only trust information within a social circle, instead of the whole social network, are applicable to all social RS algorithms outlined in the following sections.

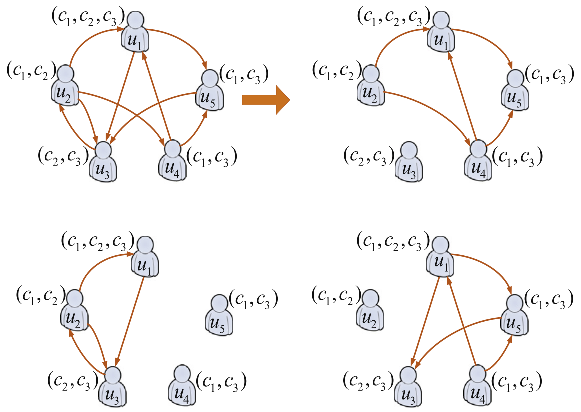

Fig. 3. Illustration of inferred circles, each user is labeled with the categories in which she has ratings. (a): the original social network; (b), (c) and (d): inferred circles for categories c_1, c_2 and c_3 respectively.

5. Matrix factorization based social recommendation approaches

The general idea of matrix factorization (MF) is to model the user-item interactions with factors representing latent characteristics of the users and items in the system, like the preference class of users and the category class of items. This model is then trained using the available data, and later used to predict ratings of users for new items.

Numerous social matrix-factorization (MF) based RSs have recently been proposed as to improve recommendation accuracy [45,46,44,47–56]. The common rationale behind all of them is that a user's taste is *similar to* and/or *influenced by* her trusted friends in the social network.

In the following subsections, we review some of the existing MF approaches in the literature that combine rating data with social network information [45,46,44,47,48]. We also review some of the nearest-neighborhood based approaches [40,53], which combine the traditional CF neighborhood with social neighborhood.

5.1. Social recommendation (SoRec) model

Social Recommendation (SoRec) was introduced in [45]. The graphical model of SoRec model is presented in Fig. 4. In SoRec, trust between users in a social network is integrated into the recommender systems by factorizing the social trust matrix S. In this model, the social network matrix S may be slightly modified as follows [45]:

$$S^*_{u,v} = S_{u,v}\sqrt{\frac{d_v^-}{d_u^+ + d_v^-}},$$

where d_u^+ is the out-degree of user u in the social network (i.e. the number of users who u follows/trusts), and d_v^- is the in-degree of user v in the network (i.e. the number of users who follow/trust user v). The predicted user-item rating-matrix is obtained from the model as follows:

$$\widehat{R} = r_m + QP^\top, \tag{12}$$

with matrices $P \in \mathbb{R}^{i_0 \times j_0}$ and $Q \in \mathbb{R}^{u_0 \times j_0}$, where $j_0 \ll i_0, u_0$ is the rank; and $r_m \in \mathbb{R}$ is a (global) offset. In addition to the rating data, also the social network information is used for training this model. The social relationships are predicted as follows:

$$\widehat{S}^* = s_m + QZ^\top, \tag{13}$$

where $Z \in \mathbb{R}^{u_0 \times j_0}$ is a third matrix in this model, besides P and Q. Note that the matrix Q is shared among the two Eqs. (12) and (13). Due to this constraint, one can expect Q (i.e., the user profile Q_u for each user u) to reflect information from both user-item ratings and user-user social trust as to achieve accurate predictions for both. Note that the matrix Z is not needed for predicting rating values, and hence may be discarded after the matrices P and Q have been learned. Both (12) and (13) are combined as follows in the training objective function to optimize RMSE:

$$\sum_{(u,i)\ \text{obs.}} \left(R_{u,i} - \widehat{R}_{u,i}\right)^2 + \sum_{(u,v)\ \text{obs.}} \left(S^*_{u,v} - \widehat{S}^*_{u,v}\right)^2 + \lambda\left(||P||_F^2 + ||Q||_F^2 + ||Z||_F^2\right), \tag{14}$$

where $||\cdot||_F$ denotes the Frobenius norm of the matrices, and λ is the usual regularization parameter. Note that obs. means observed. As to optimize ranking, the training of this model can be modified as detailed in paper [53]. Analogous to Eq. (10), the training function is modified to account for *all* items (instead of RMSE on the *observed* ratings) for an improved top-k hit-ratio on the test data:

$$\sum_{\text{all}\,u}\sum_{\text{all}\,i} W_{u,i}\left(R^{\text{o\&i}}_{u,i} - \widehat{R}_{u,i}\right)^2 + \sum_{\text{all}\,u}\sum_{\text{all}\,v} W^{(S)}_{u,v}\left(S^*_{u,v} - \widehat{S}^*_{u,v}\right)^2 + \lambda\left(||P||_F^2 + ||Q||_F^2 + ||Z||_F^2\right). \tag{15}$$

$R^{\text{o\&i}}_{u,i}$ equals the actual rating value of user u to item i if it is observed in the training data; otherwise the value $R^{\text{o\&i}}_{u,i} = r_m$ is imputed. The training weights are

$$W_{u,i} = \begin{cases} 1 & \text{if } R_{u,i} \text{ observed}, \\ w_m & \text{otherwise}. \end{cases} \tag{16}$$

The term concerning the social network is analogous to the first term concerning the ratings. In particular, the absent or unobserved social links are treated analogous to the missing ratings in AllRank [9], i.e. the value s_m with weight $w_m^{(S)}$ is imputed. Like $W_{u,i}$ in (16), $W^{(S)}_{u,v}$ is defined as follows:

$$W^{(S)}_{u,v} = \gamma \cdot \begin{cases} 1 & \text{if } S^*_{u,v} \text{ observed}, \\ w_m^{(S)} & \text{otherwise}, \end{cases} \tag{17}$$

where $\gamma \geqslant 0$ determines the weight of the social network information compared to the rating data. Obviously, $\gamma = 0$ corresponds to the extreme case where the social network is ignored when learning the matrices P and Q. As γ increases, the influence of the social network increases. The effect is that the user profiles Q_u and Q_v of two users u and v become more similar to each other if they are friends. While only positive social relationships are considered in the original model [45], we note that this model allows also for *negative* values of $S_{u,v}$, representing distrust among users. This objective function can be optimized using the popular (stochastic) gradient descent method.

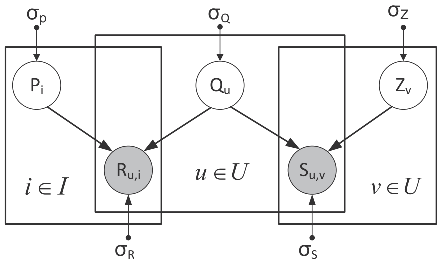

Fig. 4. The SoRec graphical model.

In paper [53], this modified training procedure was experimentally found to achieve higher top-k hit-ratios than the (modified) models outlined in the following sections. This is remarkable, because this model was found to perform rather poorly regarding the RMSE metric when compared to the models outlined in the following sections.

5.2. Social trust ensemble (STE) model

Recommendation with the Social Trust Ensemble (STE) was introduced in [46]. This approach is a linear combination of the basic matrix factorization approach [8] and a social network based approach.

The predicted ratings are obtained from a model comprising the matrices $P \in \mathbb{R}^{i_0 \times j_0}$ and $Q^{u_0 \times j_0}$:

$$\widehat{R}_{u,i} = r_m + \alpha Q_u P_i^\top + (1-\alpha) \sum_{v \in \mathcal{F}_u^+} S_{u,v} Q_v P_i^\top, \tag{18}$$

where $\mathcal{F}_u^+$ is the set of user u's direct friends. The predicted rating for item i by user u consists of three terms. The first two terms are the same as in the traditional CF: global offset r_m and prediction based on user u and item i's latent features. The last term is a weighted sum of the predicted ratings for item i from all of user u's friends. It captures the social influence. The trade-off between the feedback data (ratings) and the influence from social network is determined by $\alpha \in [0,1]$. Obviously, the social influence is ignored for $\alpha = 1$, while $\alpha = 0$ assigns the highest possible weight to the social influence. Intermediate values of α result in a weighted combination of the information from both sources. The training objective function to optimize RMSE is as follows:

$$\sum_{(u,i)\,\text{obs.}} \left(R_{u,i} - \widehat{R}_{u,i}\right)^2 + \lambda\left(||P||_F^2 + ||Q||_F^2\right). \tag{19}$$

Eq. (18) can be rewritten using matrix notation:

$$\widehat{R} = r_m + S_\alpha Q P^\top, \tag{20}$$

where $S_\alpha = \alpha I + (1-\alpha)S$, and I is the identity matrix. When the objective is ranking, in [53], the training function was modified as follows:

$$\sum_{\text{all } u} \sum_{\text{all } i} W_{u,i} \cdot \left(R_{u,i}^{\text{o\&i}} - \widehat{R}_{u,i}\right)^2 + \lambda\left(||P||_F^2 + ||Q||_F^2\right), \tag{21}$$

where $||\cdot||_F$ denotes the Frobenius norm. $W_{u,i}$ and $R_{u,i}^{\text{o\&i}}$ are defined as in the previous section. Again, this training objective function can be optimized efficiently using stochastic gradient descent.

5.3. Social MF model

Social Matrix Factorization (SocialMF) was proposed in [44], and was found to outperform SoRec and STE with respect to RMSE. The SocialMF model addresses the transitivity of trust in social networks, as the dependence of a user's feature vector on the direct neighbors' feature vectors can propagate through the network, making a user's feature vector dependent on possibly all users in the network (with decaying weights for more distant users). Each of the rows of the social network matrix S is normalized to 1, resulting in the new matrix S^* with $S_{u,v}^* \propto S_{u,v}$, and $\sum_v S_{u,v}^* = 1$ for each user u. The predicted ratings are obtained from the model, comprising the matrices $P \in \mathbb{R}^{i_0 \times j_0}$ and $Q^{u_0 \times j_0}$, as follows:

$$\widehat{R} = r_m + QP^\top. \tag{22}$$

The training objective function to optimize RMSE is as follows:

$$\sum_{(u,i)\,\text{obs.}} \left(R_{u,i} - \widehat{R}_{u,i}\right)^2 + \beta \sum_{\text{all } u} ||(Q_u - \sum_{v \in \mathcal{F}_u^+} S_{u,v}^* Q_v)||^2 + \lambda\left(||P||_F^2 + ||Q||_F^2\right), \tag{23}$$

where the second term in the objective function "forces" user u's latent feature Q_u to be similar to the (weighted) average of his/her friends' profiles Q_v (measured in terms of the square error). The tradeoff between the feedback data (ratings) and the social network information is controlled by $\beta \geqslant 0$. Obviously, the social network information is ignored for $\beta = 0$, while increasing the value of β shifts the tradeoff more and more towards the social network information.

When the goal is ranking, in [53], the training function (23) was modified as follows as to better optimize the top-k hit ratio (instead of RMSE):

$$\sum_{\text{all } u} \sum_{\text{all } i} W_{u,i} \cdot \left(R_{u,i}^{\text{o\&i}} - \widehat{R}_{u,i}\right)^2 + \beta \sum_{\text{all } u} ||(Q_u - \sum_{v \in \mathcal{F}_u^+} S_{u,v}^* Q_v)||^2 + \lambda\left(||P||_F^2 + ||Q||_F^2\right). \tag{24}$$

Also this modified training function can be optimized efficiently by means of (stochastic) gradient descent.

5.4. Similarity-based social regularization

Authors of [47] proposed social regularization as to incorporate social network information into the training procedure. They coined the term Social Regularization to represent the social constraints on recommender systems. For example, in the previous SocialMF model, the social regularization part is

$$\beta \sum_{\text{all } u} ||(Q_u - \sum_{v \in \mathcal{F}_u^+} S_{u,v}^* Q_v)||_F^2. \tag{25}$$

Authors of [47] named Eq. (25) as average-based regularization–a user's latent feature is constrained to be similar to the weighted average of whom he follows. They further proposed individual-based regularization:

$$\beta \sum_{\text{all } u} \sum_{v \in \mathcal{F}_u^+} sim(u,v) ||Q_u - Q_v||_F^2, \tag{26}$$

where a user's latent feature is constrained to be similar to his/her followees, weighted by their similarities. Similarity between users can be derived by calculating the Pearson Correlation Coefficient (PCC) or Vector Space Similarity (VSS) of commonly rated items between them. However, due to data sparsity, the number of commonly rated items between friends could be very small or even zero. To address this problem, authors of [48] improved the prediction accuracy by employing adaptive social similarities in the social regularization part. They calculate similarity between users based on their latent features. They demonstrated that latent feature based similarity function outperforms VSS and PCC similarity metric on Epinions [18] data set.

5.5. Circle-based recommendation

Similarity-based Social Regularization treats different friends differently. Circle-based recommendation further extends this idea, as only a subset of friends are taken into account when performing rating prediction in a specific circle. When applying circle-based recommendation to the SocialMF [44] model, one can build a separate MF model for each category. Using rating data only in category c and trust values in the corresponding circles $\mathcal{C}_u^{(c)}$, the training objective function to minimize RMSE becomes:

$$\mathcal{L}^{(c)}(R^{(c)}, Q^{(c)}, P^{(c)}, S^{(c)^*}) = \frac{1}{2}\sum_{(u,i)\text{obs.}}\left(R^{(c)}_{u,i} - \widehat{R}^{(c)}_{u,i}\right)^2 + \frac{\beta}{2}\sum_{\text{all } u}||(Q^{(c)}_u - \sum_v S^{(c)^*}_{u,v} Q^{(c)}_v)||^2 + \frac{\lambda}{2}\left(||P^{(c)}||^2_F + ||Q^{(c)}||^2_F\right), \quad (27)$$

where $R^{(c)}_{u,i}$ is the real rating of item i in category c, $\widehat{R}^{(c)}_{u,i}$ is the predicted rating for item i:

$$\widehat{R}^{(c)}_{u,i} = r^{(c)}_m + Q^{(c)}_u P^{(c)\top}_i, \quad (28)$$

where they define the global bias term $r^{(c)}_m$ as the average value of observed training rating in category c. The summation in Eq. (27) extends over all observed user-item pairs (u, i) in category c. Note that this model only captures user and item profiles in category c, i.e., $Q^{(c)}$ and $P^{(c)}$. $P^{(c)} \in \mathbb{R}^{i^{(c)}_0 \times j_0}$, where $i^{(c)}_0$ is the number of items in category c and $Q^{(c)} \in \mathbb{R}^{u_0 \times j_0}$.

As an alternative training objective function, one can also use rating data from all categories, instead of only the ratings in category c. The only difference from Eq. (27) is that the first line is replaced by

$$\frac{1}{2}\sum_{(u,i)\text{obs.}}\left(R_{u,i} - \widehat{R}_{u,i}\right)^2, \quad (29)$$

where the summation extends over all observed user-item pairs (u, i) from all categories. They train a separate model for each category c, i.e., $Q^{(c)}$ and $P^{(c)}$, with $P^{(c)} \in \mathbb{R}^{i_0 \times j_0}$, and $Q^{(c)} \in \mathbb{R}^{u_0 \times j_0}$. The difference from Eq. (28) is to substitute $r^{(c)}_m$ by r_m, which is the average value of all observed ratings in the training set.

6. Neighborhood based social recommendation approaches

Neighborhood based approaches use the stored ratings directly in the prediction/recommendation. We first review some social network traversal (SNT) based approaches which traverse the source-user's neighborhood in the social network and query the rating of the target item. Then, we review some of the nearest-neighborhood based approaches [40,53], which combine the traditional CF neighborhood with social neighborhood.

6.1. Social network traversal based approaches

Given a social network, some RS algorithms predict a user's rating for an item by traversing the user's neighborhood and querying the item ratings of her direct and indirect friends. We call them Social Network Traversal (SNT) based approaches.

6.1.1. Trust weighted prediction

Trust has recently been identified as an effective means to utilize social network information as to improve recommendation accuracy. Empirical studies in [28,29] found a correlation between trust and user similarity. Various techniques have been proposed to incorporate trust into CF approaches [26,27,30–37]. For instance, [30] attempts to address the rating sparsity issue using the trust relationship. It was shown that even simple binary trust relations can increase the coverage and thus the number of recommendations that can be made. [33] investigates the use of trust to better cluster users, thus improving recommendation accuracy. Typically, the rating similarity between friends is quantified by a numerical value, with larger values indicating higher levels of trust. Then recommendations are calculated for a user as a function of the ratings and the associated trust values of his friends.

MoleTrust [32] is such a SNT approach. Users are connected in a trust network, where the trust relationship is explicitly issued by users. MoleTrust considers all raters up to a *maximum-depth* given as input. *Maximum-depth* is independent of any specific user and item. Also, to compute the trust value between indirectly connected users u and v in MoleTrust, backward exploration is performed. The trust value from u to v is the aggregation of trust values between user u and users directly trusting v weighted by the direct trust values:

$$S_{u,v} = \frac{\sum_{w \in \mathcal{F}^-_v} S_{w,v} S_{u,w}}{\sum_{w \in \mathcal{F}^-_v} S_{w,v}}. \quad (30)$$

Only users within *maximum-depth*, and which have rated the target item, are considered. We denote this set by T. The rating prediction for target user u of item i in MoleTrust [32] is calculated as:

$$\widehat{R}_{u,i} = \overline{R}_u + \frac{\sum_{v \in T} S_{u,v}(R_{v,i} - \overline{R}_v)}{\sum_{v \in T} S_{u,v}}, \quad (31)$$

where $S_{u,v}$ is user u's trust of user v, and $\overline{R}_u$ is user u's average rating.

Golbeck designed a trust metric called TidalTrust [27], working in a breadth-first search fashion. TidalTrust works as follows: First, the system searches for raters that the source-user knows directly. If there is no direct connection from the source to any rater, the system moves one step out to find connections from the source to raters that are two hops away. This process repeats until a path is found. The opinions of all raters at that depth are considered. Second, using TidalTrust, the trust value is calculated for each rater at the given depth. As to infer the trust value of user u to v, who are not directly connected, TidalTrust aggregates the trust value from u's direct neighbors to v, weighted by the direct trust values from u to its direct neighbors:

$$S_{u,v} = \frac{\sum_{w \in \mathcal{F}^+_u} S_{u,w} S_{w,v}}{\sum_{w \in \mathcal{F}^+_u} S_{u,w}}. \quad (32)$$

Once the raters have been selected, the rating prediction is calculated as the weighted average of all raters' ratings:

$$\widehat{R}_{u,i} = \frac{\sum_{v \in T} S_{u,v} R_{v,i}}{\sum_{v \in T} S_{u,v}}, \quad (33)$$

where T is the set of raters within maximum-depth.

6.1.2. Bayesian inference based prediction

Different from trust-based recommendation, authors of [38] proposed to use conditional probability distributions to capture the similarity between friends in social networks. Probability distributions carry richer information than trust values, and allow one to employ Bayesian networks to conduct multiple-hop recommendation in online social networks.

In [38], each pair of friends $(u; v)$ measures their rating similarity by a set of conditional distributions $p(u|v)$ and $p(v|u)$, each of which is one user's rating distribution given the other user's ratings. $p(u|v)$ is calculated by taking out the commonly rated items between user u and v, then given user v's rating value, we calculate the rating distribution of user u on the commonly rated items. When a user wants a recommendation rating for an item, he sends out a rating query to his direct friends in the social network. Upon receiving a query for an item, a user returns its rating if he has rated the item before; otherwise the query is relayed to her friends. As to avoid loops, a user only responds to the first query from a requester, and ignores the following queries relayed through other paths. A query will be dropped after a pre-defined number of hops as to limit the range of query flooding. Fig. 5 depicts the recommendation Bayesian tree that consists of three types of users: (i) query initiator S at the root, (ii) recommenders $\{L_i\}$, who have rated the item and respond to the query with their ratings. They are leaves in the tree; (iii) intermediate

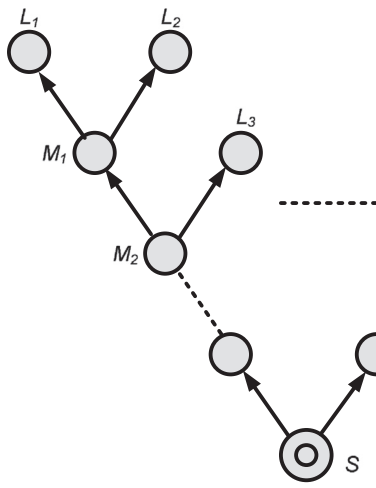

Fig. 5. Recommendation Propagation Tree as a Bayesian network.

users $\{M_j\}$, who forward and aggregate queries and responses between the initiator and the raters. Specifically, an intermediate user collects the recommendation information from his children, aggregates the information according to Bayesian calculation, and relays the aggregated information to his parent. Finally, given the structure of the recommendation propagation tree and the ratings on all leaf users, the querying initiator computes the final recommendation rating.

6.1.3. Random walk based approaches

Some other social RS algorithms employ random walks in online social networks as to compute recommendation ratings [39–41]. Authors of [39] proposed the so-called TrustWalker, which performs a random walk in online social networks as to query a user's direct and indirect friends' ratings for the target item as well as similar items. Since both ratings from similar users and ratings of similar items are considered, TrustWalker is a combination of the trust-based approach and item-item similarity based approach. Item-item similarity can be calculated using user rating information or item content information. Specifically, TrustWalker consists of two major components: random walk in the trust network and probabilistic item rating selection on each visited node. During the random walk, a user's direct and indirect friends are visited in the trust network. Whenever a friend is visited, if she has rated the target item, her rating is logged; if she has not rated the target item, but has rated an item similar to the target item, her rating is logged with certain probability. The probability of using a rating of a similar item in place of a rating for the target item increases as the length of random walk increases. This probabilistic item rating selection aims to avoid going too deep in the network when no user in a close neighborhood has rated the target item.

They employ the Pearson Correlation Coefficient of ratings expressed for two items to calculate the similarity value between them,

$$corr(i,j) = \frac{\sum_{u \in U_C}(R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{u \in U_C}(R_{u,i} - \overline{R}_u)^2 \sum_{u \in U_C}(R_{u,j} - \overline{R}_u)^2}}, \tag{34}$$

where U_C is the set of users who have rated both i and j, $R_{u,i}$ and $R_{u,j}$ are ratings of u assigned to items i and j respectively. $\overline{R}_u$ is the average rating issued by user u. Values of the Pearson correlation are in the range of $[-1, 1]$. Only items with positive correlation with the target item are considered. The similarity value is then calculated as:

$$sim(i,j) = \frac{1}{1 + e^{-\frac{|U_c|}{2}}} \times corr(i,j), \tag{35}$$

where $|U_c|$ is the number of users who rated both i and j. TrustWalker improves the prediction precision by preferring raters within a shorter distance and improves the coverage by considering ratings for similar items in addition to the target item.

The same authors extended TrustWalker to recommend top-k items for a source user u in [40]. Starting from user u, a random walk is performed in the trust network. Each random walk stops at a certain user. Then the items rated highly by that user will be considered as the recommended items, ordered according to the ratings expressed by that user. Several random walks are performed to gather more information and compute a more confident recommendation rating. The estimated rating of each item is the average of ratings for that item over all sampled raters. At the end, items with the highest estimated ratings are chosen as top-k recommended items.

6.2. Nearest neighbor (NN) methods

A NN based algorithm works by identifying the so-called neighbors of a source-user, a prediction of item preferences or a list of recommended items for him or her can be produced. In CF-based social RSs, a *NN* approach combines the traditional CF neighborhood with social neighborhood.

Authors of [40] proposed an approach, namely Trust-CF, to incorporate social network into Nearest Neighbor (NN) based top-k recommender systems. In Trust-CF, Breadth First Search (BFS), starting from a source user u, is performed to traverse the social network multiple times to obtain a set of trusted neighbors, namely trusted neighborhood. Meanwhile, it constructs a collaborative filtering (CF) neighborhood, consisting of users who are close to the source user u in terms of the Pearson Correlation Coefficient (PCC), which is obtained by computing the Pearson Correlation Coefficient of ratings from two users on their commonly rated items. The items rated highly by users in either neighborhoods are considered to be candidates for top-k recommendation. Trust-CF calculates the predicted rating for a candidate item as the weighted average of all observed ratings in the two neighborhoods. The weight for a user in the trusted neighborhood is set to $1/d_v$, where d_v is the depth of user v from user u in the trust network. The weight for a user in the CF neighborhood is the Pearson Correlation Coefficient between this user and the source-user. If an item has predicted ratings from both neighborhoods, two predicted ratings are combined using weighted average with weights proportional to the neighborhood size for this item. Finally, Trust-CF sorts all the candidate items by their predicted ratings and recommends the top-k items to the source-user.

Authors of [53] proposed Trust-CF-ULF as to incorporate social network information into top-k recommender systems. The *Trust-CF-ULF approach* is the combination of a user latent feature space based collaborative filtering approach (CF-ULF) and a social network based approach. *CF-ULF* uses MF (i.e., *AllRank* [9]) to obtain the user latent features. The users are then clustered in the user latent feature space using the Pearson Correlation Coefficient. The k_1 users nearest to the source user u are identified. Then they find k_2 closest neighbors from the trust neighborhood which are not in the k_1 set. Later on, users in the combined neighborhood vote for their relevant items. The weight for a user in the trusted neighborhood is the same as in the Trust-CF approach. The weight for a user in the CF-ULF neighborhood is the Pearson Correlation Coefficient between this user and the source-user in the user latent feature space. Finally, Trust-CF-ULF sorts all the candidate items by their received voting values, and recommends the top-k items to the source user.

Table 3
Comparison of matrix factorization based social recommendation approaches for item rating prediction task.

Representative approaches	Prediction	
	Accuracy	Training complexity
SoRec	High	$O(u_0(\bar{r}+\bar{s})j_0K)$
STE	High	$O(u_0\bar{r}\bar{s}^2j_0K)$
SocialMF	High	$O(u_0(\bar{r}+\bar{s}^2)j_0K)$
Social Regularization	High	$O(u_0\bar{r}\bar{s}j_0K)$
CicleRec	High	$O(u_0(\bar{r}+\bar{s_c}^2c_0)j_0K)$

Table 4
Comparison of matrix factorization based social recommendation approaches for item list recommendation task.

Representative approaches	Prediction	
	Accuracy	Training complexity
SoRec	High	$O(u_0(\bar{r}+\bar{s}+j_0)j_0^2K)$
STE	High	$O(u_0(i_0\bar{s}^2j_0+\bar{r}j_0^2+j_0^3)K)$
SocialMF	High	$O((u_0(\bar{s}^2j_0+\bar{r}j_0^2+j_0^3)K)$
Social Regularization	N/A	N/A
CicleRec	N/A	N/A

7. Approach comparison

Now we make a comparison of the surveyed approaches and provide a high-level summary. The comparison is focused on model-training complexity and accuracy. For the comparison of training complexity, we only outline the complexity of model-based approaches, i.e., MF-based approaches in our paper. Model complexity is the complexity of learning the model parameters during the training step. We focus on two recommendation tasks here, one is the item rating prediction task, and the other one is the item list recommendation task. Model complexity is bound to a specific optimization method. The optimization method used for the training rating prediction models is gradient descent. The optimization method used for training the item list recommendation models is alternating least squares. The accuracy metric in the rating prediction task is RMSE/MAE and the accuracy in item list recommendation task is the top-k hit ratio.

First, we compare the MF-based approaches. The comparison of different MF-based approaches surveyed in this paper is depicted in Table 3 and Table 4 for the item rating prediction task and the item list recommendation task separately. In addition to the notations introduced in Section 3, we introduce some new notations here. $\bar{r}$ is the average number of ratings per user and $\bar{s}$ is the average number of friends per user in the social network. $\bar{s}_c$ is the average number of friends per user in a social circle, c_0 is the number of circles in the social network and K is the number of iterations needed for the training of the model to converge. Among MF-based approaches, we do not report the complexity and accuracy of Social Regularization and CircleRec in the recommendation task as there is no existing work on employing them for the top-k recommendation task. For model training complexity calculation, readers can refer to model's original papers for details.

Then we make a comparison of neighborhood based social recommendation approaches. In the item rating prediction task, accuracies of MoleTrust and TidalTrust are **Low**, and accuracies of Bayesian Inference and Random Walk are **Medium**. In the item list recommendation task, Trust-CF's accuracy is **Medium** and Trust-CF-ULF's accuracy is **High**. Among neighborhood based approaches, the accuracies of Trust-CF and Trust-CF-ULF in the item rating prediction task are not available. The accuracies of MoleTrust, Tidal-Trust, Bayesian Inference and Random Walk in the item list recommendation task are not available.

We can see that, generally, model based approaches perform well in both item rating prediction and item list recommendation tasks (if available), while neighborhood based approaches enjoy the advantage of easy implementation.

8. Conclusions

In this paper, we presented a survey of CF-based social recommender systems. We first gave a short overview of the task of recommender systems and the traditional recommendation algorithms. We then presented how social network information can be adopted by recommender systems as additional input for improved accuracy. We classify CF-based social recommender systems into two categories: matrix factorization based social recommendation approaches and neighborhood based social recommendation approaches. Both types of approaches are surveyed and compared.

Current work on social recommender systems has demonstrated the effectiveness of incorporating social network information to improve recommendation accuracy. Given the increasing popularity of online social networks, new recommendation algorithms will be needed to better mine various kinds of newly available social information. Most of the surveyed algorithms are trained and tested offline. One of the next steps will be to test and improve their performance in real online social networks, with real-time user experience feedback. Finally, privacy in online social networks has attracted more and more user awareness. Privacy-preserving social recommender systems are another interesting direction for future work.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (June 2005), http://dx.doi.org/10.1109/TKDE.2005.99.

[2] F. Ricci, L. Rokach, B. Shapira, P.B. (Eds.), Kantor Recommender Systems Handbook, first ed., 2011, XXX, 842 p. 20 illus

[3] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence 2009 (2009) 19. Article ID 421425.

[4] M. Jahrer, A. Toscher, R. Legenstein, Combining predictions for accurate recommender systems, in: Proc. of KDD '10, Washington, DC, USA, 2010, pp. 693–702.

[5] R. Keshavan, A. Montanari, S. Oh, Matrix completion from noisy entries, Journal of Machine Learning Research 11 (2010) 2057–2078.

[6] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: ACM Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–34.

[7] Y. Koren, Collaborative filtering with temporal dynamics, in: Proc. of KDD '09, Paris, France, 2009, pp. 447–456.

[8] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: NIPS 2008, vol. 20.

[9] H. Steck, Training and testing of recommender systems on data missing not at random, in: ACM Conference on Knowledge Discovery and Data Mining, 2010, pp. 713–722.

[10] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: IEEE International Conference on Data Mining (ICDM), 2008.

[11] Y. Xin, H. Steck, Multi-value probabilistic matrix factorization for IP-TV recommendations, in: ACM Conference on Recommender Systems (RecSys), 2011.

[12] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: IEEE International Conference on Data Mining (ICDM), 2008.

[13] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: KDDCup, 2007.

[14] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, in: International Conference on Machine Learning (ICML), 2007.

[15] S. Deerwester, S. Dumais, G. Furnas, R. Harshman, T. Landauer, K. Lochbaum, L. Streeter et al., Latent semantic analysis/ indexing, <http://lsa.colorado.edu/>.

[16] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.

[17] YouTube, available at <http://www.youtube.com>.

[18] Epinions, available at <http://www.epinions.com/>.

[19] NetflixPrize, available at <http://www.netflixprize.com/>.

[20] Twitter, available at <http://twitter.com>.

[21] Facebook, <http://www.facebook.com>.

[22] Google Plus, <http://plus.google.com/>.

[23] Flixster, <http://www.flixster.com>.

[24] S. Funk, Netflix update: try this at home, 2006, <http://sifter.org/simon/journal/20061211.html>.

[25] M. McPherson, L. Smith-Lovin, J.M. Cook, Birds of a Feather: homophily in social networks, Annual Review of Sociology 27 (1) (2001) 415–444, http://dx.doi.org/10.1146/annurev.soc.27.1.415.

[26] J.A. Golbeck, Computing and applying trust in web-based social networks, PhD. Dissertation, 2005.

[27] J. Golbeck, J. Hendler, FilmTrust: movie recommendations using trust in web-based social networks, in: Proceedings of the IEEE Consumer Communications and Networking Conference, Jan. 2006, pp. 282–286.

[28] C.-N. Ziegler, G. Lausen, Analyzing correlation between trust and user similarity in online communities, in: Proceedings of Second International Conference on Trust Management, 2004, pp. 251–265.

[29] C.-N. Ziegler, J. Golbeck, Investigating interactions of trust and interest similarity, Decision Support Systems 43 (2) (2007) 460–475. http://dx.doi.org/10.1016/j.dss.2006.11.003.

[30] P. Massa, B. Bhattacharjee, Using trust in recommender systems: an experimental analysis, in Proceedings of iTrust2004 International Conference, 2004, pp. 221–235.

[31] P. Massa, P. Avesani, Trust-aware recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems, 2007.

[32] P. Massa, P. Avesani, Controversial users demand local trust metrics: an experimental study on epinions.com community, in: Proceedings of the 25th American Association for Artificial Intelligence Conference, 2005.

[33] T. DuBois, J. Golbeck, J. Kleint, A. Srinivasan, Improving recommendation accuracy by clustering social neworks with trust, in: Proceedings of the ACM RecSys 2009 Workshop on Recommender Systems and the Social Web, Oct. 2009.

[34] J. O'Donovan, B. Smyth, Trust in recommender systems, in: Proceedings of the 10th International Conference on Intelligent User Interfaces, 2005.

[35] F. Walter, S. Battiston, F. Schweitzer, A model of a trust-based recommendation system on a social network, Autonomous Agents and Multi-Agent Systems 16 (1) (2007) 1573–7454.

[36] K. Sarda, P. Gupta, D. Mukherjee, S. Padhy, H. Saran, A distributed trust-based recommendation system on social network, in: Proceedings of the 10th Second IEEE Workshop on Hot Topics in Web Systems and Technologies, 2008.

[37] P. Bedi, H. Kaur, S. Marwaha, Trust based recommender system for semantic web, in: Proc. of IJCAI '07, 2007, pp. 2677–2682.

[38] X. Yang, Y. Guo Y. Liu, Bayesian-inference based recommendation in online social networks, in: IEEE Transactions on Parallel and Distributed Systems (TPDS), 2013, pp. 642–651.

[39] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: ACM Conference on Knowledge Discovery and Data Mining (KDD), 2009.

[40] M. Jamali, M. Ester, Using a trust network to improve top-n recommendation, in: ACM Conference on Recommender Systems (RecSys), 2009.

[41] M.A. Tayebi, M. Jamali, M. Ester, U. Glasser, R. Frank, CrimeWalker: a recommendation model for suspect investigation, in: RecSys '11 Proceedings of the fifth ACM conference on Recommender systems, 2011.

[42] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item-based collaborative filtering recommendation algorithms, in: WWW' 01: Proceedings of the 10th International Conference on World Wide Web, ACM, New York, NY, USA, 2001, pp. 285C–295.

[43] K. Jarvelin, J. Kekalainen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems 20 (4) (2002) 422C–446. http://doi.acm.org/10.1145/582415.582418.

[44] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation, in: Social Networks Proceedings of the 2010 ACM conference on Recommender systems(RecSys), 2010.

[45] H. Ma, H. Yang, M.R. Lyu, I. King, Sorec: social recommendation using probabilistic matrix factorization, in: International Conference on Information and Knowledge Management (CIKM), 2008.

[46] H. Ma, I. King, M.R. Lyu, Learning to recommend with social trust ensemble, in: ACM Conference on Research and Development in Information Retrieval (SIGIR), 2009.

[47] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: ACM International Conference on Web Search and Data Mining (WSDM), 2011.

[48] L. Yu, R. Pan, Z. Li, Adaptive social similarities for recommender systems, in: RecSys '11 Proceedings of the Fifth ACM Conference on Recommender Systems, 2011.

[49] F. Liu, H.J. Lee, Use of social network information to enhance collaborative filtering performance, Expert System Applications 37 (7) (2010) 4772–4778.

[50] S.-H. Yang, B. Long, A.J. Smola, N. Sadagopan, Z. Zheng, H. Zha, Like alike: joint friendship and interest propagation in social networks, in: Proceedings of the 20th International Conference on World Wide Web (www2011).

[51] J. Noel, S. Sanner, K.-N. Tran, P. Christen, L. Xie, E. Bonilla, E. Abbasnejad, N.D. Penna, New objective functions for social collaborative filtering, in: Proceedings of the 21st International Conference on World Wide Web (www2012).

[52] X. Yang, H. Steck, Y. Liu, Circle-based recommendation in online social networks, in: Proceedings of 2012 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12).

[53] X. Yang, H. Steck, Y.Guo, Y. Liu, On Top-k recommendation using social networks, in: ACM Conference on Recommender Systems (RecSys'12), 2012.

[54] Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, W. Zheng, Augmenting collaborative recommender by fusing explicit social relationships, Recommender Systems & the Social Web workshop, RecSys, 2009.

[55] V. Vasuki, N. Natarajan, Z. Lu, I. Dhillon, Affiliation recommendation using auxiliary networks, in: ACM Conference on Recommender Systems (RecSys'2010), 2010.

[56] P. Symeonidis, E. Tiakas, Y. Manolopoulos, Product recommendation and rating prediction based on multi-modal social networks, in: Proceedings of the 5th ACM Conference in Recommender Systems (RecSys'2011), Chicago, IL, 2011.

[57] S.M. McNee, J. Riedl, J.A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in: Proceeding CHI EA '06 CHI '06 Extended Abstracts on Human factors in Computing Systems.

[58] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 (1) (2004) 5–53.