Circle-based Recommendation in Online Social Networks

Xiwang Yang
ECE Department
Polytechnic Institute of NYU
Brooklyn, New York
xyang01@students.poly.edu

Harald Steck*
Bell Labs
Alcatel-Lucent
Murray Hill, New Jersey
hsteck@gmail.com

Yong Liu
ECE Department
Polytechnic Institute of NYU
Brooklyn, New York
yongliu@poly.edu

ABSTRACT

Online social network information promises to increase recommendation accuracy beyond the capabilities of purely rating/feedback-driven recommender systems (RS). As to better serve users' activities across different domains, many online social networks now support a new feature of "Friends Circles", which refines the domain-oblivious "Friends" concept. RS should also benefit from domain-specific "Trust Circles". Intuitively, a user may trust different subsets of friends regarding different domains. Unfortunately, in most existing multi-category rating datasets, a user's social connections from all categories are mixed together. This paper presents an effort to develop circle-based RS. We focus on inferring category-specific social trust circles from available rating data combined with social network data. We outline several variants of weighting friends within circles based on their inferred expertise levels. Through experiments on publicly available data, we demonstrate that the proposed circle-based recommendation models can better utilize user's social trust information, resulting in increased recommendation accuracy.

1. INTRODUCTION

Recommender Systems (RS) deal with information overload by suggesting to users the items that are potentially of their interests. Traditional collaborative filtering approaches predict users' interests by mining user rating history data [1], [2], [4], [6], [15], [22] and [23]. The increasingly popular online social networks provide additional information to enhance pure rating-based RSes. Several social-trust based RSes have recently been proposed to improve recommendation accuracy, to just name a few, [9], [10], [11], [12], [14], [16], [17], and [18]. The common rationale behind all of them is that a user's taste is *similar to* and/or *influenced by* her trusted friends in social networks. Meanwhile, another obvious fact is that users' social life, being online or offline, is intrinsically *multifaceted*. To better serve a user's activities across different domains, many online social networks now support a new feature of "Friends Circles", which refines the *domain-oblivious* "Friends" concept. Google+ is the first to introduce "Circles", the function that let users assign classmates, family members, colleagues and others to different groups. Facebook, which has long had Friend lists, also launched its "Groups" feature to assign users to groups for finer granular information sharing: a user can share different information with different groups. In Twitter, users can organize people who they follow (followees) into "lists". When a user clicks to view a list, she will see a stream of Tweets from all her followees in that list.

RSes should also benefit from domain-specific "Trust Circles". 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 Cars category while not trust v in Kids' TV Show category. Therefore, u should care less about v's ratings in Kids' TV Show category than in Cars category. Ideally, if we know users' trust circles in different categories, to predict ratings in one category, we probably should only use trust circles specific to that category. We call it *circle-based recommendation*. Unfortunately, in most existing multi-category rating datasets, a user's social connections from all categories are mixed together. So if we use all social trust information for rating prediction in a specific category, we misuse social trust information from other categories, which compromises the rating prediction accuracy. Apart from that, even if the circles were explicitly known, e.g. Circles in Google+ or Facebook, they may not correspond to particular item categories that a recommender system may be concerned with. Therefore, *inferred* circles concerning each item-category may be of value by themselves, besides the explicitly known circles.

This paper presents an effort to develop circle-based RS. We focus on inferring category-specific social trust circles from available rating data combined with social network data where social trust links across all categories are mixed together. We propose a set of algorithms to infer category-specific circles of friends and to infer the trust value on each link based on user rating activities in each category. To infer the trust value of a link in a circle, we first estimate a user's *expertise level* in a category based on the rating activities of *herself as well as all users trusting her*. We then assign to users trust values proportional to their expertise levels. The reconstructed trust circles are used to develop a low-rank matrix factorization type of RS. Through experiments on publicly available data, we demonstrate that the proposed

*Now at Netflix Inc. Work was done while at Bell Labs.

circle-based RSes can better utilize user's social trust information and achieve more accurate recommendation than the traditional matrix factorization approaches that do not use any social trust information, and the existing social-trust based RSes that use mixed social trust information across all categories.

The rest of the paper is organized as follows. Section 2 presents the related work. In Section 3, we first introduce the concept of trust circle, then propose three variants of assigning weights to users within each circle. Finally, we present circle-based training models, based on either ratings from one category or ratings from all categories. Experimental results are presented in Section 4. The paper is concluded in Section 5.

2. RELATED WORK

In this paper, we focus on low-rank matrix factorization models, as they were found to be one of the most accurate single models for collaborative filtering [5,7,8,19,20]. In the following, we briefly review the ones relevant to this paper.

2.1 Matrix Factorization (MF)

While there are various sophisticated approaches (e.g. [5, 7, 8, 19, 20]), we here briefly review the basic low-rank matrix factorization (MF) approach, which will be extended towards social network information in the remainder of this paper. The matrix of predicted ratings $\hat{R} \in \mathbb{R}^{u_0 \times i_0}$, where u_0 denotes the number of users, and i_0 the number of items, is modeled as:

$$\hat{R} = r_m + QP^\top, \tag{1}$$

with matrices $P \in \mathbb{R}^{i_0 \times d}$ and $Q \in \mathbb{R}^{u_0 \times d}$, where d is the rank (or dimension of the latent space), with $d \ll i_0, u_0$, and $r_m \in \mathbb{R}$ is a (global) offset value.

This model is trained on the observed rating data by minimizing the square error (with the usual Frobenius/L2-norm regularization) (see also [5, 19]):

$$\frac{1}{2} \sum_{(u,i)\text{obs.}} \left(R_{u,i} - \hat{R}_{u,i} \right)^2 + \frac{\lambda}{2} \left(||P||_F^2 + ||Q||_F^2 \right), \tag{2}$$

where $\hat{R}_{u,i}$ denotes the ratings predicted by the model in Eq. (1); and $R_{u,i}$ are the actual rating values in the training data for item i from user u. This objective function can be minimized efficiently using gradient descent method [12].

Once the low-rank matrices P and Q have been learned, rating values can be predicted according to Eq. (1) for any user-item pair (u, i).

2.2 MF and Social Networks

The usage of social network data has been found to improve the prediction accuracy of rating values, and various models for integrating these two data sources have been proposed, like Social Recommendation (SoRec) [10], Social Trust Ensemble (STE) [9], Recommender Systems with Social Regularization [11], Adaptive social similarities for recommender systems [13], among which the SocialMF model [12] was found to achieve a particularly low RMSE value, and is hence used as a baseline model in our experimental comparison study.

2.2.1 SocialMF Model

The SocialMF model was proposed in [12], and was found to outperform SoRec and STE with respect to RMSE. The social network information is represented by a matrix $S \in \mathbb{R}^{u_0 \times u_0}$, where u_0 is the number 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 $s_m = 0$. 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 idea underlying SocialMF is that neighbors in the social network may have similar interests. This similarity is enforced by the second term in the objective function in equation (3), which says that user profile Q_u should be similar to the (weighted) average of his/her friends' profiles Q_v (measured in terms of the square error):

$$\begin{aligned} & \frac{1}{2} \sum_{(i,u) observed} \left(R_{u,i} - \hat{R}_{u,i} \right)^2 \\ + \quad & \frac{\beta}{2} \sum_{\text{all } u} \left((Q_u - \sum_v S^*_{u,v} Q_v)(Q_u - \sum_v S^*_{u,v} Q_v)^\top \right) \\ + \quad & \frac{\lambda}{2} \left(||P||_F^2 + ||Q||_F^2 \right), \end{aligned} \tag{3}$$

where the ratings $\hat{R}_{u,i}$ predicted by this model are obtained according to Eq. (1). Note that we omitted the logistic function from the original publication [12], as we found its effect rather negligible in our experiments. The trade-off between the feedback data (ratings) and the social network information is determined by a weight $\beta \geq 0$. Obviously, the social network information is ignored if $\beta = 0$, and increasing β shifts the trade-off more and more towards the social network information.

Eq. (3) can be optimized by the gradient descent approach (see the update equations (13) and (14) in [12]).

Once the model is trained, the soft constraint that neighbors should have similar user profiles is captured in the user latent feature matrix Q. The rating value for any user concerning any item can be predicted according to Eq. (1).

3. CIRCLE-BASED RECOMMENDATION MODELS

Our proposed Circle-based Recommendation (CircleCon) models may be viewed as an extension of the SocialMF model [12] to social networks with *inferred circles* of friends.

3.1 Trust Circle Inference

We 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. For instance, the circle of friends concerning cars may differ significantly from the circle regarding kids' TV shows.

To this end, we divide the 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:*

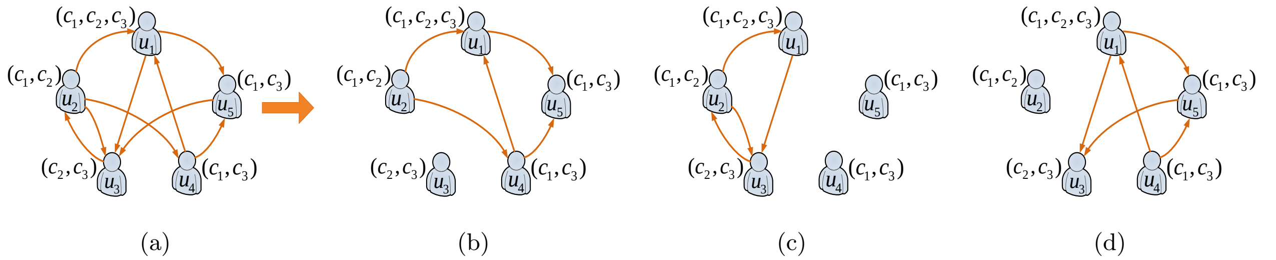

Figure 1: Illustration of inferred circles, each user is labeled with the categories in which she has ratings. a): the original social network; b), c) and d): inferred circles for categories c_1, c_2 and c_3 respectively.

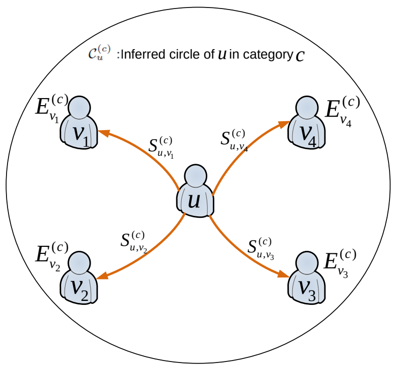

Figure 2: Illustration of expertise-based trust-assignment in category c.

- *$S_{u,v} > 0$ 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 for a toy example in Figure 1.

3.2 Trust Value Assignment

The trust values between friends in the same inferred circle (based on item category c) are captured in a social network matrix $S^{(c)}$, such that $S_{u,v}^{(c)} = 0$ if $v \notin \mathcal{C}_u^{(c)}$, $S_{u,v}^{(c)} > 0$ if $v \in \mathcal{C}_u^{(c)}$. In the following, we consider three variants of defining the positive values $S_{u,v}^{(c)} > 0$ when user v is in the inferred circle of user u regarding category c. They are then experimentally evaluated in Section 4.

3.2.1 CircleCon1: Equal Trust

We start with the simplest variant of defining trust values $S_{u,v}^{(c)} > 0$ within inferred circles regarding item category c: each user v in the inferred circle of user u gets assigned the same trust value, i.e., $S_{u,v}^{(c)*} = \text{const}$ if $v \in \mathcal{C}_u^{(c)}$. The constant is determined by the normalization constraint $\sum_{v\in\mathcal{C}_u^c} S_{u,v}^{(c)*} = 1$. In other words, $S_{u,v}^{(c)*} = 1/|\mathcal{C}_u^{(c)}|, \forall v \in \mathcal{C}_u^{(c)}$.

3.2.2 CircleCon2: Expertise-based Trust

In this section, we outline two variants of assigning *different* trust values to friends within a trust circle. The goal is to assign a higher trust value or weight to the friends that are experts in the circle / category. As an approximation to their level of expertise, we use the numbers of ratings they assigned to items in the category. The idea is that an expert in a category may have rated more items in that category than users who are not experts in that category.

We formalize this as follows. We consider directed trust relationships; undirected/mutual trust relationships (e.g., friendship) can be viewed as a special case of directed trust. If user u trusts user v in category c, we say u follows v in category c, i.e., u is the follower of v, and v is a followee of u. All of user u's followees in category c form the trust circle $\mathcal{C}_u^{(c)}$ of u in c. We also denote u's followers in category c as $\mathcal{F}_u^{(c)}$. Finally, a user u's expertise level in category c is denoted as $E_u^{(c)}$. This is illustrated in Figure 2.

We assign trust values to u's followees in circle $\mathcal{C}_u^{(c)}$to be proportional to their expertise levels in category c. Based on this idea, we consider two variants in the following:

- **Variant a:** In this case, user $v's$ expertise level in category c is equal to the number of ratings that v assigned in category c, i.e., $E_v^{(c)} = N_v^{(c)}$. Thus,

$$S_{u,v}^{(c)} = \begin{cases} N_v^{(c)} & \text{if } v \in \mathcal{C}_u^c \\ 0 & \text{otherwise.} \end{cases}$$

We then normalize each row of $S^{(c)}$ matrix as follows

$$S_{u,v}^{(c)*} = \frac{S_{u,v}^{(c)}}{\sum_{v\in\mathcal{C}_u^{(c)}} S_{u,v}^{(c)}}, \tag{4}$$

which ensures that, for each user u, the weights across all users v in each circle are normalized to unity:

$$\sum_{v\in\mathcal{C}_u^{(c)}} S_{u,v}^{(c)*} = 1.$$

- **Variant b:** In this case, the expertise level of user v in category c is the product of two components: the first component is the number of ratings that v assigned in category c, the second component is some voting value in category c from all her followers in $\mathcal{F}_v^{(c)}$. The intuition is that if most of v's followers have lots of ratings in category c, and they all trust v, it is a good indication that v is an expert in category c.

We denote the voting value from followers of v in category c by $\phi_v^{(c)}$. For each follower $w \in \mathcal{F}_v^{(c)}$, we compute the distribution of her ratings in each individual category. We denote $\mathcal{D}_w$ as a distribution vector over all the categories,

$$\mathcal{D}_w = \left(\frac{N_w(1)}{N_w}, \frac{N_w(2)}{N_w}, ..., \frac{N_w(m)}{N_w}\right), \tag{5}$$

where m is the number of categories, and $N_w(c)$ with $c = 1, ..., m$ is the number of ratings assigned by user w in category c; N_w is the total number of ratings assigned by user w, $N_w = \sum_c N_w(c)$. Thus, $\mathcal{D}_w$ records the proportions of ratings user w assigned in all categories. It reflects the interest distribution of w cross all categories.

The second component, namely the voting value from all followers is defined as $\phi_v^{(c)} = \sum_{w \in \mathcal{F}_v^{(c)}} \mathcal{D}_w(c)$.

Combining both components, we have the following expression for v's expertise level:

$$E_v^{(c)} = N_v^{(c)} \cdot \sum_{w \in \mathcal{F}_v^{(c)}} \mathcal{D}_w(c)$$

which results in the trust values

$$S_{u,v}^{(c)} = \begin{cases} N_v^{(c)} \cdot \sum_{w \in \mathcal{F}_v^{(c)}} \mathcal{D}_w(c) & \text{if } v \in \mathcal{C}_u^{(c)} \\ 0 & \text{otherwise.} \end{cases}$$

As in each of the above cases, also here we finally normalize each row of the $S^{(c)}$ matrix (across v):

$$S_{u,v}^{(c)*} = S_{u,v}^{(c)} / \sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)}.$$

3.2.3 CircleCon3: Trust Splitting

The previous circle inference and trust value assignment essentially assume that if u issues a trust statement towards v, and u and v simultaneously have ratings in a category c, then u trusts v in c. The trust value assignment is done in each circle separately. In practice, user u might issue a trust statement towards v just because of v's ratings in a subset of categories in which they simultaneously have ratings. The trust value of u towards v in category c should reflect the likelihood that u issues the trust statement towards v due to v's ratings in c. One simple heuristic is to make the likelihood proportional to the number of v's ratings in category c. In other words, given that u trusts v, if v has more ratings in category c_1 than in c_2, it is more likely that u trusts v because of v's ratings in c_1 than v's ratings in c_2. Now if u and v simultaneously have ratings in multiple categories, the trust value of u towards v should be *split* cross those commonly rated categories.

Essentially, we now normalize trust values across c,

$$S_{u,v}^{(c)} = \begin{cases} \frac{N_v^{(c)}}{\sum_{c: v \in \mathcal{C}_u^{(c)}} N_v^{(c)}} & \text{if } v \in \mathcal{C}_u^{(c)} \\ 0 & \text{otherwise.} \end{cases}$$

To illustrate this *trust splitting*, let us look at Figure 1: user u_2 trusts user u_1 and both of them have ratings in category c_1 and c_2. Assume the number of ratings u_1 issued in category c_1 and c_2 are 9 and 1 respectively. The trust value in original social network is $S_{u_2,u_1} = 1$. Now after *trust splitting*, we get $S_{u_2,u_1}^{(c_1)} = 0.9$ and $S_{u_2,u_1}^{(c_2)} = 0.1$.

Like before, we then also normalize each row of $S^{(c)}$ matrix (across v), as to make the trust values independent of the activity levels of the users in each circle:

$$S_{u,v}^{(c)*} = S_{u,v}^{(c)} / \sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)}.$$

We note that the normalizations across c and then v may also be viewed as the first step of an iterative procedure called *iterative proportional fitting* [24]. In short, when this procedure is iterated until convergence, it results in an exact *joint* normalization regarding both c and v: $\sum_c \sum_v S_{u,v}^{(c)} =$ *const*, where $\sum_c S_{u,v}^{(c)} = 1$ for each v, and $\sum_v S_{u,v}^{(c)} =$ *const* for each c. While the iterative procedure yields exact normalization, it is computationally expensive. For the latter reason, the reported results in our experiment section are obtained after only one iteration.

3.3 Model Training

3.3.1 Training with ratings from each category

Using the (normalized) trust network $S^{(c)*}$, as defined above, we train a separate matrix factorization model for each category c. For each kind of inferred circles of friends, we obtain a separate user profile $Q^{(c)}$ and item profile $P^{(c)}$ for each c. Similar to the SocialMF model [12], but with the crucial difference of using inferred social circles of friends, we use the following training objective function

$$\begin{aligned} & \mathcal{L}^{(c)}(R^{(c)}, Q^{(c)}, P^{(c)}, S^{(c)*}) = \\ & \frac{1}{2} \sum_{(u,i) \text{obs.}} \left(R_{u,i}^{(c)} - \hat{R}_{u,i}^{(c)} \right)^2 \\ + \; & \frac{\beta}{2} \sum_{\text{all } u} \left((Q_u^{(c)} - \sum_v S_{u,v}^{(c)*} Q_v^{(c)})(Q_u^{(c)} - \sum_v S_{u,v}^{(c)*} Q_v^{(c)})^\top \right) \\ + \; & \frac{\lambda}{2} \left(||P^{(c)}||_F^2 + ||Q^{(c)}||_F^2 \right), \end{aligned} \tag{6}$$

where we only use ratings $R_{u,i}^{(c)}$ in category c; $\hat{R}_{u,i}^{(c)}$ is the predicted rating of item i in category c,

$$\hat{R}_{u,i}^{(c)} = r_m^{(c)} + Q_u^{(c)} {P_i^{(c)}}^\top, \tag{7}$$

where we define the global bias term $r_m^{(c)}$ as the average value of observed training rating in category c (see also Table 4). The summation in Eq. (6) extends over all observed user-item pairs (u, i) where item i belongs to category c. Note that this model only captures user and item profiles in category c, i.e., $Q^{(c)}$ and $P^{(c)}$. $P^{(c)} \in \mathbb{R}^{i_0^{(c)} \times d}$, where $i_0^{(c)}$ is the number of items in category c and $Q^{(c)} \in \mathbb{R}^{u_0 \times d}$. Eq. (6) can be minimized by the gradient decent approach, analogous to [12]:

$$\begin{aligned} & \frac{\partial \mathcal{L}^{(c)}}{\partial Q_u^{(c)}} = \\ & \sum_{i: cat(i) = c} I_{u,i}^{R^{(c)}} \left(r_m^{(c)} + Q_u^{(c)} {P_i^{(c)}}^T - R_{u,i}^{(c)} \right) P_i^{(c)} + \lambda Q_u^{(c)} \\ + \; & \beta \left(Q_u^{(c)} - \sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)*} Q_v^{(c)} \right) \\ - \; & \beta \sum_{v: u \in \mathcal{C}_v^{(c)}} S_{v,u}^{(c)*} \left(Q_v^{(c)} - \sum_{w \in \mathcal{C}_v^{(c)}} S_{v,w}^{(c)*} Q_w^{(c)} \right), \end{aligned} \tag{8}$$

where $cat(i)$ is the category of item i.

Table 1: Epinions Data: Top-10 Category Statistics.

Category	User Count	Item Count	Rating Count	Sparsity	Trust Fraction	Original Degree	Degree in Circle
Videos & DVDs	17,312	10,065	94,261	0.999459	44.64%	29.4	13.1
Books	11,296	21,662	47,889	0.999804	36.09%	45.1	16.3
Music	10,188	14,905	43,079	0.999716	21.48%	50.0	10.7
Video Games	9,124	2,389	29,661	0.998639	13.32%	55.8	7.43
Toys	6,373	3,344	26,789	0.998743	21.49%	79.9	17.2
Online Stores& Services	8,074	973	22,661	0.997115	28.55%	63.0	18.0
Software	8,290	1,624	19,400	0.998559	22.05%	61.4	13.5
Destinations	7,438	1475	19,395	0.998232	23.12%	68.4	15.8
Cars	10,847	3,108	17,604	0.999478	19.35%	46.9	9.1
Kids' TV Shows	4,784	259	11,203	0.990958	10.85%	106.4	11.6

$$\frac{\partial \mathcal{L}^{(c)}}{\partial P_i^{(c)}} = \sum_{all\ u} I_{u,i}^{R^{(c)}} \left(r_m^{(c)} + Q_u^{(c)} {P_i^{(c)}}^T - R_{u,i}^{(c)} \right) Q_u^{(c)} + \lambda P_i^{(c)}, \tag{9}$$

where $I_{u,i}^{R^{(c)}}$ is the indicator function that is equal to 1 if u has rated i in category c, and equal to 0 otherwise. The initial values of $Q^{(c)}$ and $P^{(c)}$ are sampled from the normal distribution with zero mean. In each iteration, $Q^{(c)}$ and $P^{(c)}$ are updated based on the latent variables from the previous iteration.

Once $Q^{(c)}$ and $P^{(c)}$ are learned for each category c, this model can be used to predict ratings for user-item pairs (u, i) according to Eq. (7), where the category c of item i determines the matrices $Q_u^{(c)}$ and $P_i^{(c)}$ to be used.

3.3.2 Training with ratings for all categories.

As an alternative training objective function, we also considered using *all* ratings in the data, instead of only the ratings in category c. The only difference to Eq. (6) is that the first line is replaced by

$$\frac{1}{2} \sum_{(u,i)\text{obs.}} \left(R_{u,i} - \hat{R}_{u,i} \right)^2, \tag{10}$$

where the summation extends over all observed user-item pairs (u, i) from all categories. As before, we train a separate model for each category c, i.e., $Q^{(c)}$ and $P^{(c)}$, with $P^{(c)} \in \mathbb{R}^{i_0 \times d}$, and $Q^{(c)} \in \mathbb{R}^{u_0 \times d}$.

4. EXPERIMENTS

In this section, we evaluate our different variants of Circle-based recommendation and compare them to the existing approaches using the Epinions dataset[1].

4.1 Dataset

Epinions is a consumer opinion website where users can review items (such as cars, movies, books, software,...) and also assign them numeric ratings in the range of 1 (min) to 5 (max). Users can also express their *trust* to other users, such as reviewers whose reviews and ratings they have consistently found to be valuable. Each user has a list of trusted users. A user issues a trust statement to another user by adding the user to her trust list. In the Epinions dataset, the trust values between users are binary: if user B is in user A's trust list, then user A's trust value towards B is 1, otherwise it is 0.

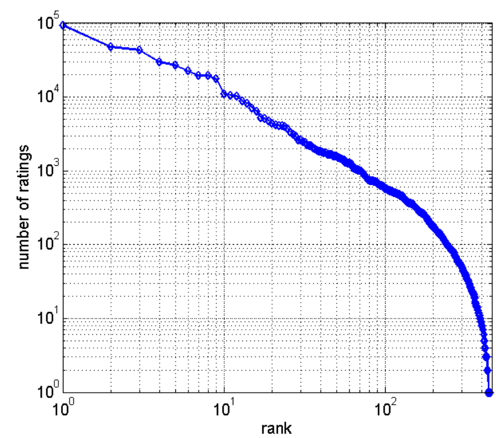

Figure 7: Distribution of Number of Ratings across Categories.

We use the version of the Epinions dataset[2] published by the authors of [3]. It consists of ratings from $71,002$ users who rated a total of $104,356$ different items from 451 categories. The total number of ratings is $571,235$. The distribution of the ratings cross all categories is plotted in Figure 7. A large number of ratings fall into a small number of large categories. The distribution of users and items in the top-10 categories is presented in Table 1.

The total number of issued trust statements is $508,960$. We apply the circle inference algorithms presented in Section 3.1 to the Epinions dataset. The fraction of trust links with $S_{u,v}^{(c)} > 0$ for each category c, out of all trust links in the entire original social network is shown in Table 1 column *Trust Fraction*. We can see that the inferred social network of each category is much smaller than the original one. A pair of users connected by a trust link in the original social

[1]http://www.epinions.com/

[2]http://alchemy.cs.washington.edu/data/epinions/

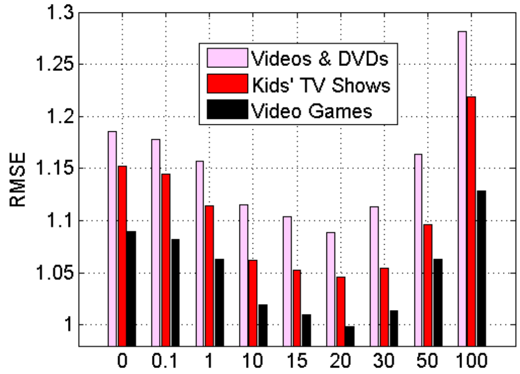

Figure 3: Impact of social information weight β on the RMSE performance using all category ratings in CircleCon3.

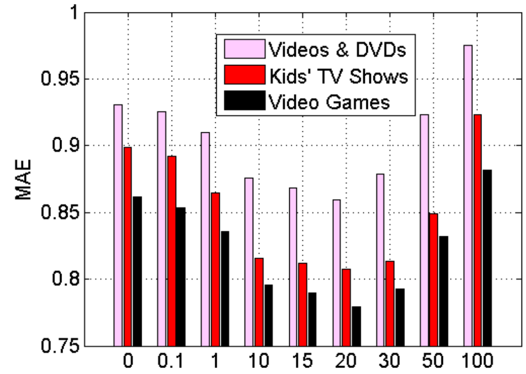

Figure 4: Impact of social information weight β on the MAE performance using all category ratings in CircleCon3.

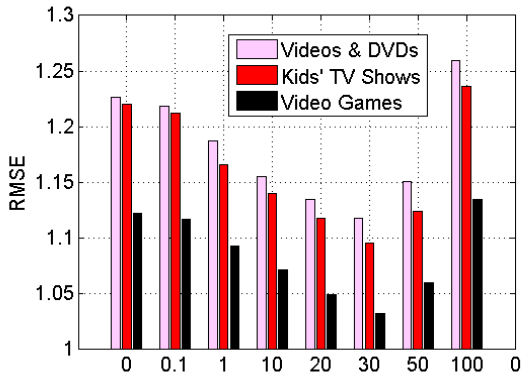

Figure 5: Impact of social information weight β on the RMSE performance using one category ratings in CircleCon3.

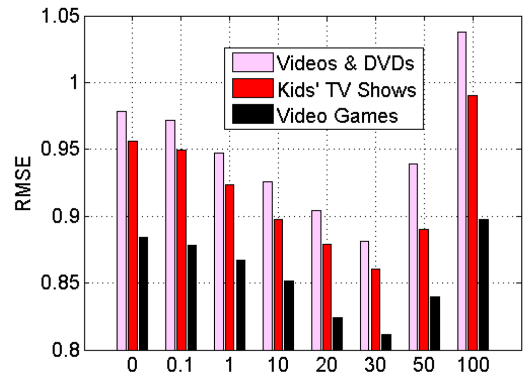

Figure 6: Impact of social information weight β on the MAE performance using one category ratings in CircleCon3.

network are not always in the same inferred circle. For instance, recommendations in the Videos & DVDs category are based on only about half of a user's friends on average, while in the Kids' TV Shows category only about 11% of friends are relevant on average. For users who have ratings in each category, we also compare their followee numbers in the original social network with those in the inferred circles for that category. The average number of followees in the original social network is shown in Table 1 column *Original Degree*. The average number of followees in the inferred circles is shown in Table 1 column *Degree in Circle*.

4.2 Performance Measures

We perform 5-fold cross validation in our experiments. In each fold, we use 80% of data as the training set and the remaining 20% as the test set. The evaluation metrics we use in our experiments are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), as these are the most popular accuracy measures in the literature of recommender systems. RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in\mathcal{R}_{test}} (R_{u,i} - \hat{R}_{u,i})^2}{|\mathcal{R}_{test}|}}, \qquad (11)$$

where $\mathcal{R}_{test}$ is the set of all user-item pairs (u, i) in the test set. MAE is defined as

$$MAE = \frac{\sum_{(u,i)\in\mathcal{R}_{test}} |R_{u,i} - \hat{R}_{u,i}|}{|\mathcal{R}_{test}|}. \qquad (12)$$

4.3 Evaluation

As to demonstrate the effectiveness of the proposed circle reconstruction approaches, we compare the recommendation results of the following approaches:

- **BaseMF**: This method is the baseline matrix factorization approach proposed in [5] and [21], which does not take into account the social network.
- **SocialMF**: This method is proposed in [12]. It improves the recommendation accuracy of BaseMF by taking into social trust between users. It always uses all social links available in the dataset.
- **CircleCon1**: For recommendation in one category c, a social link will be used if and only if the pair of users connected by the link both have ratings in category c. The social trust values are calculated as outlined in Section 3.2.1.
- **∼CircleCon1**: This is the complementary case of CircleCon1. It uses all social links except for the links

Table 2: *RMSE* comparisons for ten largest categories using all ratings (dimensionality $d = 10$). The percentage numbers in each cell are the relative improvements of CircleCon3 over the various baseline models. The standard deviations of the results are about 0.005.

Category	BaseMF	SocialMF	∼CircleCon1	CircleCon1	CircleCon2a	CircleCon2b	CircleCon3
Videos & DVDs	1.186 8.15%	1.130 3.65%	1.147 5.04%	1.098	1.109	**1.091**	**1.089**
Books	1.054 5.88%	1.023 3.09%	1.031 3.76%	1.002	1.000	**0.991**	**0.992**
Music	1.038 6.22%	0.990 1.75%	1.002 2.92%	0.979	0.982	**0.973**	**0.973**
Video Games	1.090 8.36%	1.036 3.63%	1.057 5.50%	1.005	**0.999**	**1.000**	**0.999**
Toys	1.084 9.18%	1.025 3.94%	1.031 4.56%	0.990	0.995	**0.988**	**0.984**
Online Stores & Services	1.329 11.05%	1.236 4.31%	1.257 5.95%	1.189	**1.182**	1.190	**1.182**
Software	1.153 7.54%	1.093 2.50%	1.107 3.76%	1.072	1.075	**1.061**	**1.066**
Destinations	1.032 7.49%	0.984 3.03%	0.993 3.85%	0.961	**0.959**	0.963	**0.955**
Cars	1.123 4.66%	1.085 1.35%	1.094 2.12%	1.076	1.080	**1.073**	**1.070**
Kids' TV Shows	1.153 9.28%	1.083 3.48%	1.095 4.50%	1.052	1.051	**1.042**	**1.046**

Table 3: *MAE* comparison for ten largest categories using all ratings (dimensionality = 10). The percentage numbers in each cell are the relative improvements of CircleCon3 over the various baseline models. The standard deviations of the results are about 0.005.

Category	BaseMF	SocialMF	∼CircleCon1	CircleCon1	CircleCon2a	CircleCon2b	CircleCon3
Videos & DVDs	0.931 7.66%	0.890 3.44%	0.901 4.54%	**0.863**	0.874	**0.862**	**0.860**
Books	0.862 7.36%	0.834 4.23%	0.842 5.16%	0.812	0.809	**0.797**	**0.799**
Music	0.849 7.14%	0.806 2.27%	0.811 2.80%	0.794	0.798	**0.788**	**0.788**
Video Games	0.861 9.49%	0.809 3.67%	0.824 5.35%	0.786	**0.781**	**0.781**	**0.780**
Toys	0.859 10.31%	0.808 4.69%	0.814 5.38%	0.777	0.782	**0.773**	**0.770**
Online Stores & Services	1.050 9.78%	0.994 4.77%	1.008 6.03%	0.955	**0.947**	0.953	**0.947**
Software	0.910 8.23%	0.858 2.64%	0.869 3.93%	0.841	0.842	**0.831**	**0.835**
Destinations	0.838 9.40%	0.795 4.44%	0.802 5.37%	0.766	0.766	0.769	**0.759**
Cars	0.867 4.86%	0.834 1.08%	0.841 1.90%	0.832	**0.828**	**0.829**	**0.825**
Kids' TV Shows	0.899 10.13%	0.846 4.49%	0.852 5.23%	0.813	0.812	**0.802**	0.808

used by CircleCon1. The trust value assignment is same as **CircleCon1**.

- **CircleCon2**: It uses the same subset of social links used by **CircleCon1**. But the social trust values are calculated according to the two expertise-based normalization algorithms presented in Section 3.2.2, resulting into two variations: **CircleCon2a** and **CircleCon2b**.
- **CircleCon3**: It uses the same subset of social links used by **CircleCon1**. But the social trust values are calculated according to the trust-splitting algorithm presented in Section 3.2.3.

Table 4: Performance comparison when training on *ratings of the given category only.* The standard deviations of the results are about 0.005.

Category	avg. rating	BaseMF		SocialMF		CircleCon3	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
Videos & DVDs	3.77	1.227 8.84%	0.978 9.89%	1.160 3.59%	0.921 4.28%	**1.118**	**0.882**
Books	4.30	1.065 5.79%	0.869 5.90%	1.029 2.47%	0.841 2.80%	**1.003**	**0.818**
Music	4.30	1.055 7.25%	0.855 6.72%	0.999 2.06%	0.812 1.69%	**0.979**	**0.798**
Video Games	4.06	1.122 8.04%	0.884 8.20%	1.074 3.92%	0.844 3.85%	**1.032**	**0.812**
Toys	4.14	1.137 11.12%	0.887 9.89%	1.063 4.90%	0.832 3.90%	**1.011**	**0.799**
Online Stores & Services	3.52	1.446 15.28%	1.172 15.99%	1.277 4.07%	1.034 4.75%	**1.225**	**0.985**
Software	4.06	1.202 8.49%	0.955 9.11%	1.157 4.95%	0.916 5.18%	**1.099**	**0.868**
Destinations	4.38	1.049 7.01%	0.851 7.81%	0.992 1.62%	0.804 2.44%	**0.975**	**0.785**
Cars	4.17	1.140 5.33%	0.871 4.94%	1.099 1.82%	0.845 2.06%	**1.079**	**0.828**
Kids' TV Shows	4.14	1.221 10.23%	0.956 9.98%	1.154 5.04%	0.904 4.84%	**1.096**	**0.860**

In all our experiments, we set the dimensionality for low-rank matrix factorization to be $d = 10$ and the regularization constant to be $\lambda = 0.1$.

4.3.1 Training with ratings from all categories

In this section, we consider the variant of using all ratings for training the various models, as outlined in Section 3.3.2. The other variant, which uses only the ratings of a single category at a time, will be evaluated in the next section.

In detail, regarding **BaseMF**, we use all ratings from all categories as input to train for the latent features of all items and all users; for **SocialMF**, we use all category ratings and all trust links as input to train for the latent features of all items and all users. For all the circle-based methods, we conduct separate trainings for individual categories, as discussed in Section 3.3.2.

When trained with all ratings, for **SocialMF**, the optimal social information weight is $\beta = 15$. For **CircleCon1**, **CircleCon2** and **CircleCon3**, the optimal social information weights are all around $\beta = 20$. We plot the RMSE and MAE of **CircleCon3** as a function of β in Figures 3 and 4, respectively. The trends for **CircleCon1** and **CircleCon2** are similar. This is because after we filtered out the trust links which do not belong to the current category, the similarity between a follower and followee connected by a surviving trust link should be higher. In ~CircleCon1, we set $\beta = 15$.

The performance of the five methods are compared in Tables 2 and 3 regarding RMSE and MAE, respectively. The standard deviations of the results are about 0.005. Values that are within the standard deviation of the best result are highlighted in bold. We can see from the tables that **CircleCon1** is better than **SocialMF**, and **~CircleCon1** is worse than **SocialMF**, in terms of both RMSE and MAE. This demonstrates from two sides the benefit of reconstructing trust circles for individual categories. Over all categories, **CircleCon3** has the best performance, but **CircleCon2b** is slightly (but not significantly) better than **CircleCon3** in the categories of Books, Software and Kids' TV Shows.

4.3.2 Training with per-category ratings.

As outlined in Section 3.3.1, another training alternative is to use only the ratings pertaining to the category that the model is trained for. In detail, for **BaseMF**, we only use ratings in the current category as input. For **SocialMF**, we use ratings in the current category and all trust links as input. For circle-based RSes, as discussed in 3.3.1, we use ratings in the current category and weighted trust links obtained from circle construction methods. Among the different variants, we only presents results for **CircleCon3** in this section, as we found it to be the best approach in the previous section.

When trained with per-category ratings, for **SocialMF** the optimal social information weight is $\beta = 15$. As illustrated in Figures 5 and 6, for **CircleCon3**, the optimal social information weight is $\beta = 30$. Similar to the case of training with all ratings, this is again because the latent features between friends in circles established for each category should be more similar to each other. The detailed results of per-category training are shown in Table 4.

We can see from Table 4, **CircleCon3** is better than **SocialMF** in terms of RMSE and MAE.

4.4 Observations

Comparing the performance of **BaseMF** in Table 2 and Table 4 (also Table 3 and Table 4), we can see that more rating information enables BaseMF to learn user's interest more accurately. The same conclusion holds when comparing the performance of **SocialMF** in Table 2 and Table 4 (also Table 3 and Table 4). Comparing **CircleCon3** in Table 2 and Table 4 (also Table 3 and Table 4), we can see that all category ratings combined with per-category Circle trust relationships provides the best prediction accuracy in all the compared cases. This is because all category ratings provide us more information to learn user's interests, and circles reconstructed for individual categories alleviate the ambiguity resulting from the mixed trust statements issued cross different categories.

Thus, we find that the proposed variant **CircleCon3** with all category ratings achieves the lowest RMSE and MAE values overall.

5. CONCLUSIONS

In this paper, we presented a novel approach to improving recommendation accuracy by introducing the concept of "inferred circles of friends". The idea is to determine the best subset of a user's friends, i.e., an inferred circle, for making recommendations in an item category of interest. As these inferred circles are tailored towards the various item categories, they may differ from explicit circles of friends that have recently become popular in online social networks. We proposed methods for inferring category-specific circles, and to assign weights to the friends within each circle. In our experiments on publicly available data, we showed significant improvements over existing approaches that use mixed social network information.

6. ACKNOWLEDGMENTS

The authors would like to thank Debasis Mitra and Shiv Panwar for initiating this collaboration. Steck would like to thank Tin Kam Ho for her encouragement and support of this work.

7. REFERENCES

[1] G. Adomavicius and 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*, 17:734–49, 2005.

[2] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In Proc. of KDD '07, pages 95-104, San Jose, California, USA, 2007.

[3] M. Richardson and P. Domingos. Mining Knowledge-Sharing Sites for Viral Marketing, In Proc. of KDD, 2002.

[4] M. Jahrer, A. Toscher, and R. Legenstein. Combining predictions for accurate recommender systems. In Proc. of KDD '10.

[5] S. Funk. Netflix update: Try this at home, 2006. `http://sifter.org/ simon/journal/20061211.html`.

[6] Y. Zhang, B. Cao, and D. Y. Yeung. Multi-domain collaborative filtering. In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI), Catalina Island, California, USA, 2010.

[7] R. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–78, 2010.

[8] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proc. KDD, 2008.

[9] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *ACM conference on Research and development in information retrieval (SIGIR)*, 2009.

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

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender Systems with Social Regularization. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.

[12] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[13] L. Yu, R. Pan, and Z Li. Adaptive social similarities for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* , 2011.

[14] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In Proc. of IJCAI '07, pages 2677-2682, 2007.

[15] Y. Koren. Collaborative filtering with temporal dynamics. In Proc. of KDD '09, pages 447-456, Paris, France, 2009.

[16] F. Liu and H. J. Lee. Use of social network information to enhance collaborative filtering performance. Expert Syst. Appl., 37(7):4772-4778, 2010.

[17] P. Massa and P. Avesani. Trust-aware recommender systems. In Proc. of RecSys '07, pages 17-24, Minneapolis, MN, USA, 2007.

[18] J. O 'Donovan and B. Smyth. Trust in recommender systems. In Proc. of IUI '05, pages 167-174, San Diego, California, USA, 2005.

[19] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDDCup*, 2007.

[20] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *International Conference on Machine Learning (ICML)*, 2007.

[21] R. Salakhutdinov and A. Mnih Probabilistic matrix factorization. In NIPS 2008, volume 20.

[22] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In Proc. of SIGIR '06, Seattle, Washington, USA, 2006.

[23] G -R Xue, C Lin, Q Yang, W Xi, H -J Zeng, Y Yu,and Z Chen. Scalable collaborative filtering using cluster-based smoothing. In Proc. of SIGIR '05, pages 114-121, Salvador, Brazil, 2005.

[24] W. Deming and F. Stephan, On least square adjustment of sampled frequency tables when the expected marginal totals are known, *Ann. Math. Statist.*, 6, pp. 427-44, 1940.