SoCo: A Social Network Aided Context-Aware Recommender System

Xin Liu
École Polytechnique Fédérale de Lausanne
Batiment BC, Station 14
1015 Lausanne, Switzerland
x.liu@epfl.ch

Karl Aberer
École Polytechnique Fédérale de Lausanne
Batiment BC, Station 14
1015 Lausanne, Switzerland
karl.aberer@epfl.ch

ABSTRACT

Contexts and social network information have been proven to be valuable information for building accurate recommender system. However, to the best of our knowledge, no existing works systematically combine diverse types of such information to further improve recommendation quality. In this paper, we propose SoCo, a novel context-aware recommender system incorporating elaborately processed social network information. We handle contextual information by applying random decision trees to partition the original user-item-rating matrix such that the ratings with similar contexts are grouped. Matrix factorization is then employed to predict missing preference of a user for an item using the partitioned matrix. In order to incorporate social network information, we introduce an additional social regularization term to the matrix factorization objective function to infer a user's preference for an item by learning opinions from his/her friends who are expected to share similar tastes. A context-aware version of Pearson Correlation Coefficient is proposed to measure user similarity. Real datasets based experiments show that SoCo improves the performance (in terms of root mean square error) of the state-of-the-art context-aware recommender system and social recommendation model by 15.7% and 12.2% respectively.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; J.4 [**Computer Applications**]: Social and Behavior Sciences

Keywords

Recommender System, Context-awareness, Matrix Factorization, Social Networks

1. INTRODUCTION

By suggesting information (from a huge volume of information pool) that is likely to interest users, recommender systems have become a promising tool to handle information overload in many online application scenarios like e-commerce (e.g., Amazon, Netflix), social networks (e.g., LinkedIn, Fouthsquare) and review sites (e.g., Movielens, Epinions), to name a few [23]. Most recommender systems rely on collaborative filtering techniques [2, 25], which predict a user's interest in an item by mining the patterns from the past rating information of other *similar* users and/or items. Although collaborative filtering has become the standard method for the recommendation problem, traditional recommender systems only use ratings of relevant users/items to make recommendations without taking into account any other information. This trait, when information volume becomes larger and larger, poses crucial challenges such as data sparsity (i.e., insufficient similar users/items can be found), low recommendation quality (due to data sparsity, as well as homogeneity of the information source), etc.

In order to handle the issues of traditional recommendation models, recently, two trends in the community of recommender systems have attracted a lot of attention: (1) Contextual information (e.g., time, location, mood, weather, etc.) has been recognized as an important factor that influences the accuracy of recommendations. For instance, Bill may prefer watching action films with his brothers, but would rather choose a romance film with his girlfriend. In this case, the companions (brothers vs. girlfriend) is the key contextual information for movie recommendation. Several context-aware recommender systems have been proposed [5, 22, 30] to incorporate contextual information into existing recommendation frameworks, e.g., matrix factorization models [13]. (2) The fast growth of online social networks has brought another trend of so called social[1] recommendation which relies on the opinions of the target user's friends who are assumed to share similar interests [18, 19, 28]. In theory, social recommendation can help to mitigate the issues of data sparsity (i.e., a user's preference for an unrated item can be inferred from his/her friends) and recommendation quality (i.e., friends are normally have similar preference for the same items).

However, existing context-aware recommender systems either cannot efficiently combine different types of contextual information (e.g., the contexts with discrete values versus the ones with continuous values [11]) or suffer from high computational complexity (e.g., matrix factorization model is impractical for extremely large dataset, or multiple matrix factorization operations are needed [30]). More impor-

[1]In some papers, social relationships are considered to be a type of contextual information. However, given their dynamics and complexity, it is non-trivial to combine social relationships with other contextual information such as time, location, etc. So in this work, we decouple social relationships from other contexts and process this type of information differently.

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

tantly, to the best of our knowledge, no social network based recommender system systematically incorporates rich, diverse types of contexts, which are essential to make personalized and accurate recommendations[2]. For instance, some recent attempts only consider group-aware friendship [28, 27], or only specific contextual information such as time or user mood when an item is rated is considered for social recommendation [16, 17]. Therefore, a more sophisticated recommendation mechanism (than the state-of-the-art approaches) that is able to systematically and efficiently combine different types of information (i.e., contextual information and social relationships) to further improve recommendation quality is desired.

In this paper, we propose SoCo, a novel context-aware recommender system incorporating elaborately processed social network information. The main contributions of SoCo are summarized as follows: (1) We first extract diverse contextual information which is expected to be associated with user preference. Then we apply random decision trees algorithm to partition the given user-item-rating matrix (see Fig. 1(a) as an example) taking into account various contextual information (i.e., features that are associated with each node of a decision tree). The generated sub-matrices (i.e., leaves of a decision tree) contain 'similar' ratings[3] which impose higher impacts on each other. In other words, the missing ratings inferred from the sub-matrices are more accurate and personalized than those directly derived using the original rating matrix. Note that since random decision trees induction is applied, SoCo is able to integrate diverse types of contextual information that may have different value domains (e.g., discrete value versus continuous value). (2) We employ matrix factorization model [13], which is one of the most successful approaches for recommendation, to predict missing preference of a user for an item using the generated sub-matrix. On the basis of a matrix factorization model, we introduce an additional social regularization term to improve recommendation quality considering the influence of a user's friends. Instead of employing all available social information, we select friends who share similar tastes with the target user by investigating their past ratings. A context-aware Pearson Correlation Coefficient is proposed to measure user similarity. (3) We conduct experiments on two real datasets to demonstrate SoCo's performance. Experimental results show that, in terms of Root Mean Square Error (RMSE), SoCo improves the performance of basic matrix factorization model, social recommendation model and context-aware recommender systems by 25.4%, 12.2% and 15.7% respectively.

The rest of this paper is organized as follows: In Section 2, we provide background information about matrix factorization model and review related context-aware and social recommendation models. In Section 3, we present SoCo, a social network aided context-aware recommender system. Specifically, in Section 3.1, the recommendation problem is formalized along with notation definition. Then random decision trees based context-aware matrix partition mechanism is described in Section 3.2, followed by an enhancement leveraging social network information in Section 3.3. We report real datasets based experimental results in Section 4. In Section 5 we conclude this paper and outline future research directions.

[2]In [4], the authors proposed a system architecture for personalized recommendation considering both social networks and contexts. However, the paper does not reveal any details about how different types of information are combined from an algorithmic perspective.

[3]This kind of similarity is determined by various contextual information, i.e., two ratings are similar means they are generated in the similar contexts.

2. BACKGROUND AND RELATED WORKS

Before presenting SoCo, we first provide background knowledge about matrix factorization and discuss how a matrix factorization model can be applied to predict a user's preference for an item. Then, we review related works from two areas: (1) context-aware recommender systems that integrate contextual information to improve recommendation quality, and (2) social recommender systems that rely on social network information, i.e., social recommendation.

2.1 Matrix factorization

Basically, the goal of matrix factorization is to factorize a matrix into two (or more) matrices such that by multiplying the factorized matrices, the original matrix can be reconstructed or approximated. In the context of recommendation problem, a matrix factorization model factorizes a user-item-rating matrix $R \in \mathcal{R}^{m\times n}$ (m is the number of users and n is the number of items) into one user-specific matrix $U \in \mathcal{R}^{l\times m}$ and one item-specific matrix $V \in \mathcal{R}^{l\times n}$:

$$R \approx U^T V. \tag{1}$$

where l is the dimension of a latent factor vector which characterizes a user or an item. For a user a, the elements of U (i.e., U_a) measure a's interest in items which have high values on the corresponding latent factors; for an item b, the elements of V (i.e., V_b) measure the strength of correlation between b and the corresponding latent factors. Accordingly, the resulting $U_a^T V_b$ captures the correlation between user a and item b, i.e., a's preference for b, taking into account all latent factors.

In order to approximate R, the following objective function is defined, considering sparseness of the user-item-rating matrix (i.e., a huge portion of rating values in R are missing):

$$\arg\min_{U,V} \sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2. \tag{2}$$

where I_{ij} is 1 if user i has rated item j, and 0 otherwise. Furthermore, in order to avoid overfitting, a regularization term is added to the equation:

$$\arg\min_{U,V} \sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 + \lambda(\| U \|_F^2 + \| V \|_F^2). \tag{3}$$

where $\| A \|_F^2$ (A is a $X \times Y$ matrix) is the Frobenius norm, calculated by $\sqrt{\sum_x^X \sum_y^Y |A_{xy}|^2}$. The parameter λ controls the extent of regularization.

Equation 3 can be solved (i.e., minimized) using two approaches: (i) stochastic gradient descent (SGD), which iteratively updates user-specific latent factors and item-specific latent factors [9] and (ii) alternating least squares (ALS) that fixes U (or V) and optimizes V (or U), and then rotates, iteratively [32].

2.2 Context-aware recommender systems

Contextual information has proved to be useful for providing more accurate prediction in various application domains [21] including recommender systems. Contexts can be obtained in several ways, such as by explicitly gathering from relevant users/items, by implicitly deriving from data or environment, or by inferring using statistical methods, or data mining/machine learning, etc. [3]. Adomavicius et al. [1] presented a multidimensional recommendation model based on multiple dimensions, i.e., user/item dimension as well as various contextual information. Before being used, contextual information is preprocessed by utilizing various statistical tests such that only the contexts that are truly impactful are chosen for recommendation [27].

Recent works have focused on building models that directly integrate contextual information with traditional user-item-rating relations. For instance, Karatzoglou et al. [11] proposed a multiverse recommendation model by modeling the data as a user-item-context N-dimensional tensor. Tucker decomposition is applied to factorize the tensor [26]. However, this model is only applicable for categorical contextual information. A further improvement was proposed to cater to all types of contexts [22]. However, although the authors claims that the proposed model is able to bring down computational complexity, if the original user-item-rating matrix is huge, the model may still suffer from scalability issue. One possible solution to alleviate scalability issue is to partition the original matrix before applying any factorization models.

Zhong et al. [30] proposed a contextual collaborative filtering algorithm (called RPMF) to support context-aware recommendation. The assumption behind this model is that contextual information is encoded in or reflected by the user-specific and item-specific latent factors. Based on this, tree based random partition is applied to split the user-item-rating matrix by grouping users and items with similar contexts, and then apply matrix factorization to the generated sub-matrices. Although our proposed SoCo employs a similar tree based method, they are still significantly different: (1) RPMF implicitly handles contextual information by dealing with values of latent factors (i.e., the contexts are assumed to be embedded in latent factors), while SoCo explicitly processes contexts. (2) RPMF applies matrix factorization to each node of a tree, while SoCo only works on the leaf node (i.e., only one matrix factorization operation for each tree). Another key point that makes RPMF and SoCo different is that RPMF does not take into account any social network information. We will compare SoCo with RPMF in the evaluation section to demonstrate the advantage of the way we handle contextual information.

2.3 Social recommendation

Using social network information to recommend items has become another hot topic in the area of recommender systems. In [18], the authors proposed probabilistic matrix factorization based approach to fuse user-item-rating matrix and users' social network information. In [14], a neighborhood-based approach is developed to generate social recommendations. A set of experiments were conducted to compare social based and nearest neighbor based recommendations. Ma et al. [19] introduced the social regularization on the basis of matrix factorization to constrain the taste difference between a user and his/her friends. Two variants are proposed: (1) average-based regularization that targets to minimize the difference between a user's latent factors and average of that of his/her friends; (2) individual-based regularization that focuses on latent factor difference between a user and each of his/her friends. This work also compared the performance of different similarity measures, i.e., Vector Space Similarity and Pearson Correlation Coefficient.

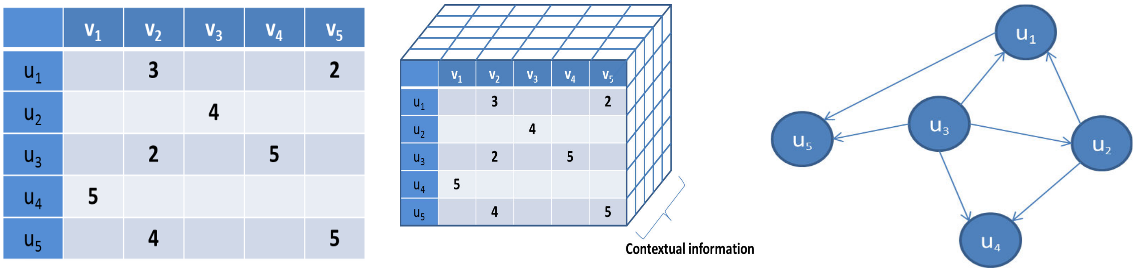
However, most existing social recommendation models largely ignore contexts when measuring similarity between two users. For instance, even if a friend has very similar tastes with a user, her rating on a movie may be greatly influenced by other factors, for instance, her mood, or with whom she watched the movie. Recent works have started looking at contexts when handling social network information. For instance, Xu et al. [27] proposed to cluster users and items such that like-minded users and their items are grouped. Subgroup information (i.e., a type of context) is then utilized by applying collaborative filtering to improve top-N recommendation quality. Yang et al. [28] first argued that a user may trust different subsets of friends regarding different domains, and then proposed a category specific circle-based model to make context-aware recommendation. However, these works only consider very basic contextual information (e.g., category/group). Akther et al. [4] proposed an architecture to collect contexts and social network information for personalized recommender systems. The authors focused on how relevant data is collected and stored but ignored how such data is efficiently combined from an algorithmic perspective. Jiang et al. [10] proposed to integrate social contexts (individual preference and interpersonal influence) into a matrix factorization model. However, such contextual information is only related to social relationships, so non-social contexts are largely ignored. In contrast, by applying machine learning techniques and matrix factorization, SoCo incorporates a variety of contextual information without the restriction on information type from two aspects: (i) contexts are explicitly considered to partition the rating matrix, (ii) a context-aware Pearson Correlation Coefficient is proposed to improve the accuracy of user similarity measure.

3. SOCO RECOMMENDER SYSTEM

In this section, we present SoCo, a social network aided context-aware recommender system. We first formalize the context-aware social recommendation problem and define notations in Section 3.1. Our context-aware recommendation approach and its social network based enhancement are elaborated in Section 3.2 and Section 3.3 respectively.

3.1 Preliminaries

Traditional recommender systems normally only consider the user-item-rating matrix to make recommendations (see Fig. 1(a)). However, in many systems, rich contextual information is available, providing a new information dimension for recommendation (see Fig. 1(b)). We classify contextual information into two categories: (1) static context, which describes characteristics of a user, e.g., age, gender, membership, role, etc. or an item, e.g., category, cost, physical properties, etc.; (2) dynamic context, which represents instantaneous information that is associated with a rating (e.g., a user's mood or location when he/she rates an item). On the other hand, online social networks have brought another information source by which a user's preference for an item can be inferred from his/her friends who are expected



(a) User-item-rating matrix. (b) Context-aware user-item-rating matrix. (c) Social network graph.

Figure 1: Context-aware social recommendation.

to share similar tastes (see Fig. 1(c)). Therefore, in this work, we endeavor to systematically integrate contextual information and social network information into a matrix factorization model to improve recommendation quality.

We denote the user set by $\mathcal{U} = \{U_1, U_2, ..., U_m\}$, and the item set by $\mathcal{V} = \{V_1, V_2, ..., V_n\}$. Any user can rate any item based on his/her preference. We assume that the value of a rating is a discrete variable in a range $\mathcal{L} = \{L_1, L_2, ..., L_l\}$. For instance, many recommender systems, e.g., MovieLens, employ five-point likert scale (e.g., [1, 2, 3, 4, 5]). A rating provided by user U_u on the item V_v is denoted by $R_{u,v}$, and all ratings $\mathcal{R} = \{R_{u,v} | U_u \in \mathcal{U}, V_v \in \mathcal{V}\}$ construct a user-item-rating matrix (see Fig. 1(a) as an example). As mentioned before, we also assume a set of contextual information that is associated with each rating R_i, denoted by $C_i = \{c_1, c_2, ...\}$. Note that all ratings have the same contextual information vector and we have no restriction on the value domain of each type of contextual information, i.e., both discrete values and continuous values are acceptable. Regarding the social network, we define a directed graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where edge set $\mathcal{E}$ represents the relations between users ($\mathcal{U}$). We denote the friend set of a user U_u by $\mathcal{F}_u \subset \mathcal{U}$.

3.2 Context-aware recommendation

We first discuss how to incorporate contextual information to improve recommendation quality without taking into account social relationships. In order to efficiently combine different contextual information, we apply a random decision trees algorithm, which is one of the most accurate learning algorithms to construct multiple decision trees randomly [8]. The rationale behind this approach is to partition the original rating set $\mathcal{R}$ (i.e., user-item-rating matrix) such that ratings with similar contexts are grouped. Since generated in similar contexts, ratings in the same cluster are expected to be better correlated among each other than those in original rating matrix (i.e., the missing ratings can be inferred more accurately).

When constructing *each* decision tree, at each level of the tree, we randomly select one contextual information c_r from the set C to partition the rating matrix (see Fig. 2 as an example). Specifically, the rating matrix is partitioned according to the value of c_r. For instance, if we assume contextual information c_r is *day-of-week*, the rating matrix can be meaningfully partitioned accordingly to which day (i.e., from Sunday to Saturday, or weekday versus weekend) the items are rated. On the other hand, if the value of c_r has no semantic meaning (e.g., a user's average rating when he/she rates a specific item), we first normalize every rating's c_r

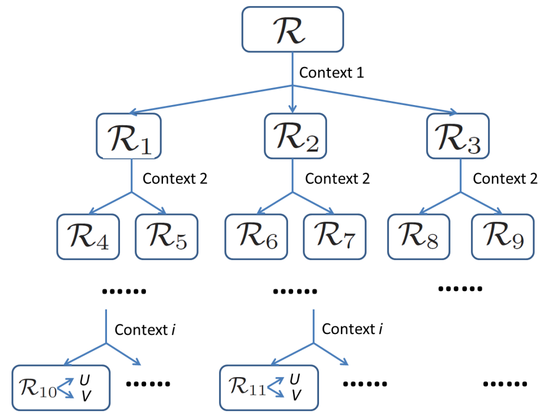


Figure 2: Random decision trees (one tree).

value into a certain range, e.g., [0,1], and then choose a random threshold value (e.g., $\in [0, 1]$) to partition the ratings. Once rating partition at one level is completed, the randomly selected contextual information c_r is removed from the contextual information set: $C = C \setminus c_r$ such that one contextual information is processed only once in a path.

Note that in a specific application scenario, any contextual information can be identified, some of which is closely correlated with a user-item-rating interaction, while some other contexts are not. It is thus important to select the most *relevant* contextual information before partitioning the ratings. Machine learning, data mining and various statistical methods [12, 15] can be applied to preprocess diverse contextual information, however, detailed discussion on context selection is out of the scope of this paper[4].

The partition process continues until one of following requirements is met: (i) all contextual information is processed; (ii) the limitation on the height of a tree has been reached; (iii) there are not a sufficient number of ratings to split at the current node. After partitioning, the ratings $\mathcal{R}$ are classified based on diverse contextual information (i.e., leaves of a tree in Fig. 2). Note that in different decision trees, the training ratings are classified differently, given that contexts (and their values) are selected randomly at each level of a tree. When predicting a missing rating R_m, we assume T decision trees, and in each decision tree, R_m is classified to the rating subset (i.e., user-item-rating sub-matrix) $\mathcal{R}_i^s \subset \mathcal{R}$ according to R_m's contextual information. For each $\mathcal{R}_i^s$ (e.g., $\mathcal{R}_{10}$ in Fig. 2), we decompose it and

[4]The work by Adomavicius et al. [1] is the first attempt to investigate the relevance/usefulness of different contexts.

get the factorized user-specific and item-specific matrices U_i^s and V_i^s, which can be used to predict the missing rating R_m (see Eq. 4 and 5):

$$\mathcal{L}_1 = \arg\min_{U_i^s, V_i^s} \sum_{j=1}^{|U_i^s|} \sum_{k=1}^{|V_i^s|} I_{jk}(\mathcal{R}_{i,j,k}^s - (U_{i,j}^s)^T V_{i,k}^s)^2 + \lambda(\| U_i^s \|_F^2 + \| V_i^s \|_F^2). \tag{4}$$

$$R_{m,i} = (U_i^s)^T V_i^s. \tag{5}$$

Finally, the predictions from T decision trees are combined to generate the final prediction for R_m.

$$R_m = \frac{\sum_{i=i}^{T} R_{m,i}}{T} \tag{6}$$

By combining multiple predictions from different decision trees, all contextual information is comprehensively investigated, generating personalized and accurate context-aware recommendations. Moreover, by removing less-relevant (in terms of context) ratings, the generated sub-matrix is significantly smaller than the original rating matrix, which is normally very huge in practice, indicating that the computational complexity is greatly reduced. One important factor that influences the complexity of our approach is the number of decision trees. We will show in the evaluation section that in real application scenarios, only a small number of trees are sufficient to achieve high quality recommendation, demonstrating practicability of our approach.

It is worth mentioning that by partitioning the rating matrix, the target user and/or the target item may have no past ratings in the generated sub-matrix (i.e., removed due to different contexts with the missing rating R_m), thus causing more cold-start issues. This can be partially solved by introducing a small amount of the removed ratings (available in the original rating matrix) that are provided by the target user and/or on the target item. Such ratings can be selected based on context similarity, i.e., the ratings that have more similar contexts with the missing rating R_m will be selected with higher priority[5]. In case the target user and/or the target item are completely new in the system, i.e., no single rating is available in the original rating matrix, the algorithms that are particularly designed for addressing cold-start issue may be applied [31], but this is beyond the scope of this paper.

3.3 An enhanced model aided by social relationships

On the basis of our context-aware recommendation model that is presented in the previous section, in this section, we describe an enhanced version of SoCo taking into account social network information to further improve recommendation quality. The assumption of this model is intuitive: in real world scenarios, when we decide whether or not to buy a product, e.g., book, CD, movie ticket, we often ask for suggestions from our friends, whose tastes are expected to be similar with ours. By combining opinions from multiple friends, we are able to make wise decisions.

[5]Note that each rating is associated with a contextual information vector (see Section 3.1), and we apply Pearson Correlation Coefficient to measure context based similarity.

Although friends' opinions provide valuable information to help in making high quality recommendation for users, most existing works either utilize/mix all available social network information without fine grained information filtering [18] or do not deeply investigate how to precisely measure taste similarity between two users [14]. In order to address these issues, following the approach proposed in [19], we introduce a new social regularization term to constrain taste difference between a user and his/her friends. In the real world, a user may have hundreds or even thousands of friends, it is thus meaningless to treat all friends (and their information) equally because some friends may have quite similar tastes with the user, while some others may have totally different tastes. In order to address such social taste heterogeneity, the introduced social regularization term takes into account taste similarity between a user and each of his/her friends:

$$\alpha \sum_{j=1} \sum_{f \in \mathcal{F}_j} S(j,f) \| U_{i,j} - U_{i,f} \|_F^2 . \tag{7}$$

where α is a constant controlling the extent of social regularization. $S(j,f)$ indicates the taste similarity between user u_j and one of his/her friends u_f based on their past rating patterns. A large similarity score means based on past commonly rated items, u_j and u_f have very similar taste, while a small similarity score means u_j and u_f's tastes are quite dissimilar.

It is worth mentioning that in different social networks, the friend relationship may be symmetric or asymmetric. We denote $\mathcal{F}_j^+$ as a set of users with whom u_j actively makes friends, and $\mathcal{F}_j^-$ as a set of users who actively make friend with u_j. In some social networks like Facebook, $\mathcal{F}_j^+$ is equal to $\mathcal{F}_j^-$, but in other social networks like Twitter, $\mathcal{F}_j^+$ and $\mathcal{F}_j^-$ are not identical. In this work, when we mention u_j's friend set $\mathcal{F}_j$, we refer to $\mathcal{F}_j^+$.

From Eq. 7 we can see that an effective way to incorporate social network information is to accurately weight each friend's opinion by investigating similarity between the user and this friend, which can be measured based on their past rating patterns, i.e., characteristics of the items that both users have commonly rated. There are many similarity calculation methods, among which Pearson Correlation Coefficient (PCC) [6] has been proven to be more accurate than other methods like vector space similarity in many scenarios [19]. So in this work, we apply PCC to measure similarity between u_j and his/her friend u_f:

$$S(j,f) = \frac{\sum\limits_{v \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} (R_{j,v} - \overline{R_j})(R_{f,v} - \overline{R_f})}{\sqrt{\sum\limits_{v \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} (R_{j,v} - \overline{R_j})^2} \cdot \sqrt{\sum\limits_{v \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} (R_{f,v} - \overline{R_f})^2}}. \tag{8}$$

where $\mathcal{V}(j) \bigcap \mathcal{V}(f)$ is the set of items that u_j and u_f have commonly rated, $\overline{R_j}$ and $\overline{R_f}$ are the average ratings of u_j and u_f respectively.

One advantage of PCC (by considering average rating) is that it takes into account the fact that some users tend to give high ratings (e.g., 4 or 5 in five-point likert scale) to most items, while some more serious users may generally issue low ratings (e.g., 2 or 3 in five-point likert scale). However, this classic similarity measure only utilizes the values of the ratings, without taking into account any contexts, which are another class of useful information for similarity estimation [20, 7]. In order to further improve the accuracy of user similarity calculation, we propose a context-aware version of Pearson Correlation Coefficient:

$$S_c(j,f) = \sum_{v \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} w_v (R_{j,v} - \overline{R_j})(R_{f,v} - \overline{R_f}) / \left(\sqrt{\sum_{v \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} w_v (R_{j,v} - \overline{R_j})^2} \sqrt{\sum_{v \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} w_v (R_{f,v} - \overline{R_f})^2} \right). \quad (9)$$

where c indicates context-awareness. The weight w_v of each item V_v is calculated by:

$$w_v = \frac{N(pcc_v)}{\sum_{v' \in \mathcal{V}(j) \bigcap \mathcal{V}(f)} N(pcc_{v'})}. \quad (10)$$

where the function $N(.)$ normalizes the given value to the range [0,1], and pcc_v represents the PCC between u_j's rating $R_{j,v}$ and u_f's rating $R_{f,v}$ on the same item V_v. Note that this PCC is measured by a contextual information vector that is associated with each rating[6] (see contextual information vector defined in Section 3.1). Clearly, large weight (w_v) means that a user and his/her friend rate the same rating in similar contexts, thus imposing high impact on overall similarity measure.

We normalize the context-aware similarity score $S_c(j, f)$ from the range [-1,1] to [0,1] before applying it to the recommendation model. Using Eq. 7 and Eq. 9, Eq. 4 is reformulated as:

$$\begin{aligned} \mathcal{L}_2 = \arg\min_{U_i^s, V_i^s} & \sum_{j=1}^{|U_i^s|} \sum_{k=1}^{|V_i^s|} I_{jk} (\mathcal{R}^s_{i,j,k} - (U^s_{i,j})^T V^s_{i,k})^2 + \\ & \alpha \sum_{j=1} \sum_{f \in \mathcal{F}_j} S_c(j,f) \parallel U_{i,j} - U_{i,f} \parallel_F^2 + \\ & \lambda(\parallel U_i^s \parallel_F^2 + \parallel V_i^s \parallel_F^2). \end{aligned} \quad (11)$$

Eq. 11 can be solved by performing gradient descent in $U^s_{i,j}$ and $V^s_{i,k}$, which are iteratively updated.

$$U^s_{i,j} \leftarrow U^s_{i,j} + \gamma \frac{\partial \mathcal{L}_2}{U^s_{i,j}}. \quad (12)$$

$$V^s_{i,k} \leftarrow V^s_{i,k} + \gamma \frac{\partial \mathcal{L}_2}{V^s_{i,k}}. \quad (13)$$

where γ is the learning rate. The influence of the number of iterations on the performance will be studied in the evaluation section.

It is worth noting that in certain cases, the generated sub-matrix may not contain sufficient opinions from a user's friends (i.e., his/her friends have few ratings that are in the similar context with the missing rating). In order to address this issue, based on context similarity, we select a set of past ratings provided by the user's friends and add them to the sub-matrix such that the recommendation model is able to benefit from social network information. We leave as a future work a more detailed discussion on selection of the friends' ratings, and the tradeoff between the useful social information and the introduced noises.

[6]Note that we only consider the contextual information that is computable for Pearson Correlation Coefficient.

4. EVALUATION

In this section, we conduct comprehensive experiments to evaluate the performance of SoCo by comparing with the state-of-the-art recommender systems.

4.1 Experimental methodology

4.1.1 Datasets

Douban[7] is one of the largest Chinese social platforms for sharing reviews and recommendations for books, movies and music. Each user can provide ratings (ranging from one star to five stars) to books, movies and music, indicating his/her preference on the item. A timestamp is associated with a rating. A user, although has not consumed an item (i.e., no rating is provided), may still express his/her interest by indicating "wish" (e.g., wish to read the book). A social network is provided, where one user can follow another user whose reviews are considered to be interesting and useful. Table 1 demonstrates the statistics of the dataset. Note that we only use explicit ratings, i.e., the "wish" expressions are not considered to be ratings.

Table 1: Statistics of the Douban dataset

	# of ratings	# of users	# of items
Book	812,037	8,598	169,982
Movie	1,336,484	5,227	48,381
Music	1,387,216	23,822	185,574
All	3,535,737	25,560	403,937

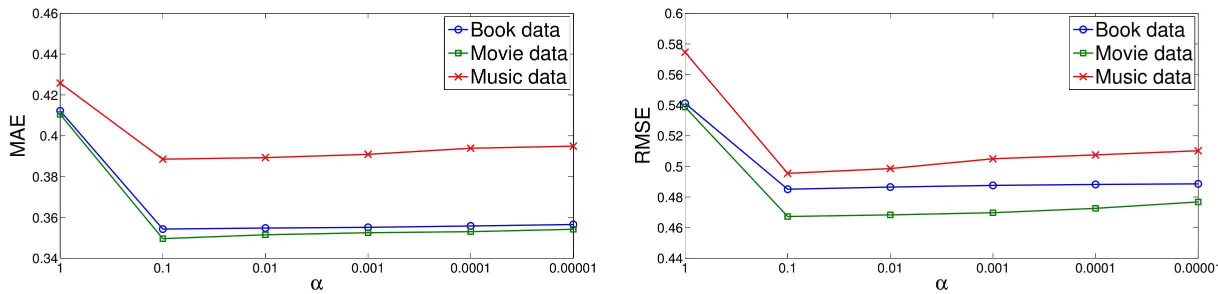
We choose the Douban data[8] because it contains not only time/date related and other inferred contextual information, but also social relationships information, thus is suitable for evaluating the performance of SoCo, which utilizes various types of information. In contrast, in some application scenarios such as MovieLens[9] and Netflix[10], social network information is not available. So in order to demonstrate the advantages of SoCo without social information, we use MovieLens-1M data[11], which is collected from a movie recommender system. The dataset consists of about 1 million ratings of approximately 3900 movies made by 6040 users. Ratings are also made on a 5-star scale, and each user has at least 20 ratings.

For both Douban and MovieLens datasets, we randomly select 80% of the ratings to train recommendation models and compare their performance using the rest 20% of the ratings.

4.1.2 Comparisons

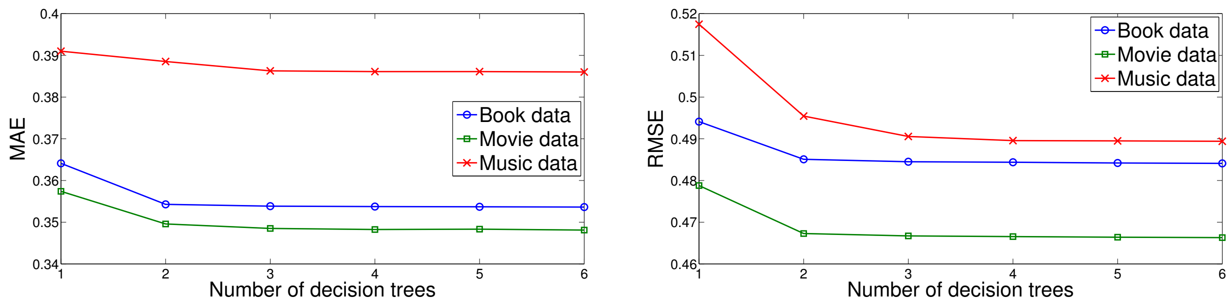
To the best of our knowledge, no existing work systematically combines contextual information and social network information to make high quality recommendations. Nevertheless, we compare SoCo with the state-of-the-art context-aware recommender system and social recommender system, as well as a basic matrix factorization model and item/user-based collaborative filtering algorithms:

[7]www.douban.com
[8]This dataset is shared by Erheng Zhang [29]
[9]http://movielens.org
[10]www.netflixprize.com
[11]http://www.grouplens.org/node/12



(a) MAE. (b) RMSE.

Figure 3: Impact of parameter α ($\lambda = 0.1$, dimensionality $= 10$, iteration # $= 20$).



(a) MAE. (b) RMSE.

Figure 4: Impact of number of decision trees ($\lambda = 0.1$, $\alpha = 0.01$ dimensionality $= 10$, iteration # $= 20$).

RPMF [30], short for *R*andom *P*artition *M*atrix *F*actorization, is a contextual collaborative filtering model based on a tree structure constructed by using random partition technique. Specifically, the ratings generated with similar contexts are partitioned onto the same node of a decision tree. Then matrix factorization is applied to the current rating matrix at each node to predict the missing ratings. Multiple predictions at different nodes and trees are combined to produce the final recommendation. Significantly different from SoCo, RPMF does not explicitly handle contextual information, but assumes it is encoded in the latent factor vector of each user/item, which means RPMF partitions the ratings based on the values of latent factors, instead of the values of real contexts (refer to Section 2.2).

SoReg [19] is a social network information based recommendation model. On the basis of a basic matrix factorization model, the authors added a new social regularization to control friends' opinions. Two variants are proposed: (1) average-based regularization, which constrains the difference between a user' taste and average of his/her friends' tastes; (2) individual-based regularization that constrains the difference between a user's taste and that of each of his/her friends individually. In the experiments, we only compare SoCo with individual-based variant which is proved to be more accurate.

BMF uses basic matrix factorization technique to predict missing ratings without considering any contextual information and social network information (see Section 2.1).

Item-based collaborative filtering algorithm [24] first finds a set of the most similar items (with the target item) that the target user has rated and then predicts the rating on the target item by taking a weighted averaging of the ratings on these similar items.

User-based collaborative filtering algorithm aggregates ratings from a set of "neighbors" who share the similar rating patterns with the user in question.

Note that for all context-aware recommender systems, based on available information from the datasets, we extract five types of contextual information: (1) hour-of-day, i.e., which hour a rating is given; (2) day-of-week, i.e., what day a rating is given; (3) number of "wish" on the target item when a rating is given (for Douban data only); (4) average value of the ratings provided by the target user when he/she rates a certain item; (5) category of the target item.

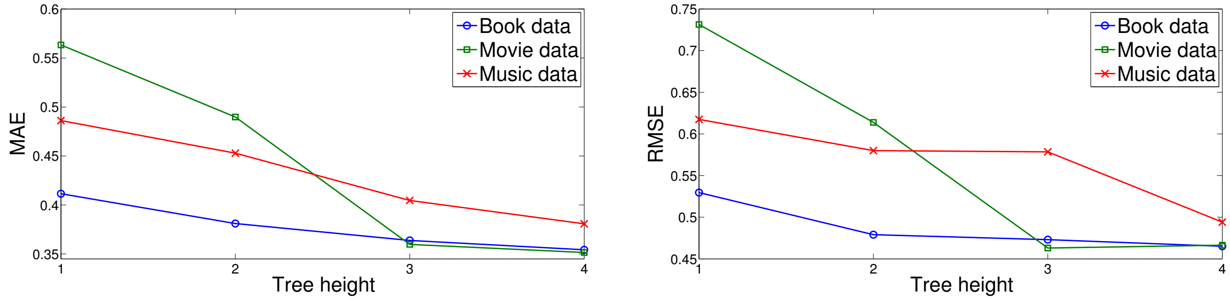
4.1.3 Metrics

We use two standard metrics to measure and compare the performance of various recommendation models: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are defined using Eq. 14 and 15 respectively:

$$MAE = \frac{1}{N}\sum_{r=1}^{N}|R_r - R_r^{'}|. \tag{14}$$

and

$$RMSE = \sqrt{\frac{1}{N}\sum_{r=1}^{N}(R_r - R_r^{'})^2}. \tag{15}$$



(a) MAE. (b) RMSE.

Figure 5: Impact of volume of contextual information ($\lambda = 0.1$, $\alpha = 0.01$ dimensionality = 10, iteration # = 20).

where N is the total number of predictions, R_r is the real rating of an item and R'_r is the corresponding predicted rating. Note that Each experiment was repeated ten times. Since low variance is observed in all experiments (i.e., 95% confidence interval), we do not show error bars to avoid cluttering. All comparison related results are statistically significant, proved by two-tailed, paired t-test with p-values $<$ 0.001.

4.2 Results

4.2.1 Performance on Douban data

We first use the Douban dataset to demonstrate the impact of various parameters of SoCo. We set the regularization constant $\lambda = 0.1$, which is determined by cross-validation. Fig. 3 shows that when different subset of the dataset (i.e., book data, movie data and music data) is applied, how the performance of SoCo varies with different values of parameter α, which controls how much social network information is incorporated into SoCo. Note that we set latent factor vector dimensionality and the number of iterations for solving a matrix factorization model to 10 and 20 respectively. We will later on show how these two parameters influence the performance of different matrix factorization based recommendation models. We observe that when α increases, both MAE and RMSE first decrease, and then become relatively stable (but slightly increase) when α reaches a certain threshold, i.e., around 0.01. We thus conclude that social network information is able to effectively improve recommendation quality and $\alpha = 0.01$ is a suitable threshold that nicely balances the user-item-rating matrix and social network information.

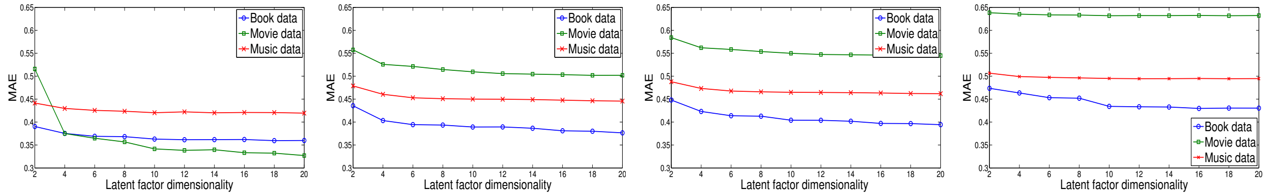
Another parameter that impacts the performance of SoCo is the number of decision trees that are employed to predict a missing rating. Fig. 4 shows that a small number of decision trees (e.g., 2 or 3) can achieve high quality of recommendation. This result also demonstrates that SoCo's computational complexity (in terms of random decision trees based rating partition) is reasonable in practice. In the following experiments, we set the number of decision trees to 3 for SoCo.

We then evaluate the impact of volume of contextual information. To do so, we control the height of the decision trees. That is, if we set the height to 1, only one type of contextual information is used in each tree; if we set the height to 4, all contextual information is employed for recommendation. From Fig. 5 we observe that in all cases, more contextual information produces higher performance, i.e., lower MAE and RMSE. This demonstrates, on one hand, contextual information greatly improves recommendation quality, on the other hand, the selected contexts (see Section 4.1.2) are quite useful.

We also conduct experiments to study the impact of the similarity function (see Eq. 8 and 9) and find that our proposed context-aware PCC reduces MAE/RMSE (compared to original PCC) by around 4.25%/5.46% on average (book data, movie data and music data).

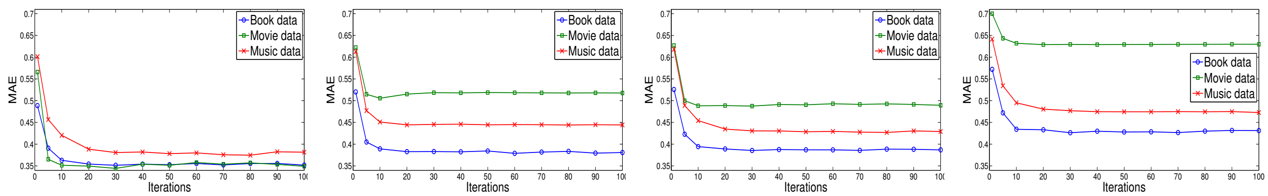
Finally, we compare the performance of SoCo with that of other recommender systems using the Douban dataset. Before comparison, we determine two important parameters, i.e., latent factor vector dimensionality and the number of iterations for matrix factorization based models. We first fix the iteration number to 10, and show MAE and RMSE with varying dimensionality of latent factor vector (see Fig. 6). We observe that MAE/RMSE decreases with increasing dimensionality, which means larger dimensionality produces higher accuracy. However, when the dimensionality increases to around 10 (even 8 for some cases), improvements on recommendation quality become negligible. We thus conclude that even a small number of latent factors are sufficient for matrix factorization based models (on the Douban data). For the following experiments, we set the latent factor vector dimensionality for SoCo, SoReg, RPMF and BMF to the threshold values that achieve stably low MAE/RMSE (i.e., [8,10]). Similarly, as shown in Fig. 7, we set the threshold numbers of iterations (i.e., [20,30]) for all matrix factorization based models because more iterations incur higher computational overheads without evidently lowering down MAE and RMSE in return.

Once the parameters are determined, we compare the performance of various recommendation models using book data, movie data, music data and entire Douban data respectively. Table 2 summarizes comparison results. We notice that in all experiment scenarios, SoCo is more accurate than other recommendation models. All matrix factorization based models significantly outperform traditional item-based and user-based collaborative filtering algorithms, demonstrating the advantage of matrix factorization technique in the area of recommender systems. These results also show that considering both contextual information and social network information provides higher recommendation quality than the models that only take into account one type of this information (i.e., SoReg and RPMF). The fact that



(a) SoCo. (b) SoReg. (c) RPMF. (d) BMF.

Figure 6: Impact of latent factor dimensionality (iteration # = 10).



(a) SoCo. (b) SoReg. (c) RPMF. (d) BMF.

Figure 7: Impact of number of iterations.

SoReg is slightly better than RPMF indicates that carefully processed social network information contributes more to a recommendation model (at least on the Douban dataset). To sum up, when the entire Douban dataset is used, in terms of MAE, SoCo improves the performance as high as 16.0%, 20.0% and 26.9% in contrast to SoReg, RPMF and BMF respectively. In terms of RMSE, the corresponding improvements are 12.2%, 15.7% and 25.4% respectively.

4.2.2 Performance on MovieLens-1M data

Experiments conducted on the Douban dataset demonstrate that combining contextual information and social network information greatly improves recommendation quality. However, in some application scenarios, no social functionality is provided (e.g., Netflix). In order to evaluate the performance of SoCo when only non-social contextual information is available, we conduct another set of experiments on MovieLens-1M dataset. Note that the social recommendation model SoReg is not involved in the comparison due to lack of essential social information.

From Fig. 8 we observe that the results share the similar trends with Douban data based experiments. User-based collaborative filter algorithm incurs highest MAE and RMSE, while item-based algorithms performs much better. All matrix factorization based models outperform traditional memory based algorithms, which again demonstrates the advantage of the latent factor models. Both SoCo and RPMF outperform BMF, proving that incorporating a variety of contextual information can achieve higher recommendation quality. The advantage of SoCo indicates that the way we structure and incorporate contextual information imposes higher impact on the improvement of recommendation quality. This is mainly because SoCo explicitly processes contextual information, but RPMF assumes that the contextual information is embedded into latent factors and partition the ratings based on latent factors instead of the contexts themselves. Moreover, the fact that RPMF runs matrix factorization at each level of each tree greatly increases computational overheads, i.e., under the same experiment setting, one running of RPMF is much (around 5×) slower than other matrix factorization based models[12]. Overall, SoCo improves the performance (MAE/RMSE) as high as 2.7%/2.6%, 7.4%/6.9%, 10.6%/11.0% and 24.2%/25.4% in contrast to RPMF, BMF, item-based and user-based collaborative filtering algorithms respectively.

5. CONCLUSION

In this paper, we propose SoCo, which systematically combines contextual information and social network information to improve quality of recommendations. SoCo first partitions the original rating matrix based on various contexts using random decision trees algorithm. The generated submatrix contains ratings with similar contexts thus imposing higher impact on each other. Matrix factorization is applied to the sub-matrix to predict the missing ratings. In order to efficiently incorporate social network information, SoCo introduces an additional social regularization term to infer a user's preference for an item by learning his/her friends' tastes. To identify friends with similar tastes, a context-aware version of Pearson Correlation Coefficient is proposed to measure user similarity. Experiments conducted on two real datasets show that SoCo evidently outperforms the state-of-the-art context-aware and social recommendation models. Moreover, even if in some scenarios where social network information is not available, SoCo still outperforms other context-aware approaches by efficiently organizing and incorporating various contextual information.

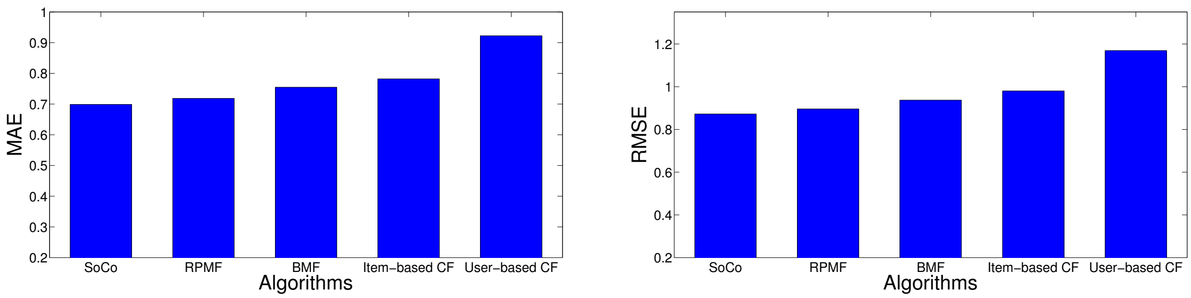
In the future work, we intend to apply SoCo to some real-world application scenarios. For instance, in our RecONCILE project[13], SoCo can be integrated into a web content credibility evaluation system where rich contextual information collected from web contents and the associated social connections can be utilized by SoCo for efficient credible web content recommendation.

[12]This is also observed in the Douban data based experiments.

[13]http://lsir.epfl.ch/research/current/reconcile/

Table 2: Performance comparison on the Douban dataset

	Book		Movie		Music		All	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SoCo	**0.3543**	**0.4651**	**0.3515**	**0.4664**	**0.3885**	**0.4954**	**0.3675**	**0.4788**
SoReg	0.3828	0.4945	0.5151	0.6416	0.4444	0.5293	0.4374	0.5451
RPMF	0.3994	0.5102	0.5526	0.6632	0.4563	0.5311	0.4594	0.5681
BMF	0.4331	0.5711	0.6288	0.8063	0.4769	0.6073	0.5029	0.6416
Item-based CF	0.9084	1.2832	0.8557	1.0544	1.1782	1.4420	0.9807	1.2598
User-based CF	1.2887	1.6535	1.0508	1.4446	1.3450	1.6890	1.2281	1.5957



(a) MAE. (b) RMSE.

Figure 8: Performance comparison on MovieLens-1M dataset.

6. ACKNOWLEDGEMENT

This work was partially supported by the grant Reconcile: Robust Online Credibility Evaluation of Web Content from Switzerland through the Swiss Contribution to the enlarged European Union.

7. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, pages 217–253. 2011.

[4] A. Akther, K. Alam, H.-N. Kim, and A. El Saddik. Social network and user context assisted personalization for recommender systems. In *Innovations in Information Technology (IIT), 2012 International Conference on*, 2012.

[5] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, 2011.

[6] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of UAI*, 1998.

[7] M. Buchin, S. Dodge, and B. Speckmann. Context-aware similarity of trajectories. In *Geographic Information Science*, volume 7478 of *Lecture Notes in Computer Science*, pages 43–56. Springer Berlin Heidelberg, 2012.

[8] R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, 2008.

[9] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011.

[10] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012.

[11] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[12] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*, 1996.

[13] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[14] F. Liu and H. J. Lee. Use of social network information to enhance collaborative filtering performance. *Expert Systems with Applications*, 37(7):4772 – 4778, 2010.

[15] H. Liu and H. Motoda. Feature selection for knowledge discovery and data mining. *Springer*, 1998.

[16] N. N. Liu, B. Cao, M. Zhao, and Q. Yang. Adapting neighborhood and matrix factorization models for context aware recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, 2010.

[17] N. N. Liu, L. He, and M. Zhao. Social temporal

collaborative ranking for context aware movie recommendation. *ACM Transactions on Intelligent Systems and Technology*, 4(1), 2013.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008.

[19] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011.

[20] D. Morent, D. Patterson, and M. Berthold. Towards context-aware similarity metrics. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, 2005.

[21] C. Palmisano, A. Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE Trans. on Knowl. and Data Eng.*, 20(11):1535–1549, 2008.

[22] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[23] F. Ricci, L. Rokach, B. Shapira, and P. Kantor. *Recommender Systems Handbook*. Springer, 2011.

[24] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, 2001.

[25] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, Jan. 2009.

[26] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

[27] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, 2012.

[28] X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012.

[29] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li. Comsoc: adaptive transfer of user behaviors over composite social network. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012.

[30] E. Zhong, W. Fan, and Q. Yang. Contextual collaborative filtering via hierarchical matrix factorization. In *Proceedings of the SIAM International Conference on Data Mining*, 2012.

[31] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[32] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management*, AAIM '08, 2008.