

HeteroMF: Recommendation in Heterogeneous Information Networks using Context Dependent Factor Models

Mohsen Jamali
Department of Computing Science
University of British Columbia
Vancouver, Canada
jamalim@cs.ubc.ca

Laks V.S. Lakshmanan
Department of Computing Science
University of British Columbia
Vancouver, Canada
laks@cs.ubc.ca

ABSTRACT

With the growing amount of information available online, recommender systems are starting to provide a viable alternative and complement to search engines, in helping users to find objects of interest. Methods based on Matrix Factorization (MF) models are the state-of-the-art in recommender systems. The input to MF is user feedback, in the form of a rating matrix. However, users can be engaged in interactions with multiple types of entities across different contexts, leading to multiple rating matrices. In other words, users can have interactions in a heterogeneous information network. Generally, in a heterogeneous network, entities from any two entity types can have interactions with a weight (rating) indicating the level of endorsement. Collective Matrix Factorization (CMF) has been proposed to address the recommendation problem in heterogeneous networks. However, a main issue with CMF is that entities share the same latent factor across different contexts. This is particularly problematic in two cases: Latent factors for entities that are cold-start in a context will be learnt mainly based on the data from other contexts where these entities are not cold-start, and therefore the factors are not properly learned for the cold-start context. Also, if a context has more data compared to another context, then the dominant context will dominate the learning process for the latent factors for entities shared in these two contexts. In this paper, we propose a context-dependent matrix factorization model, HETEROMF, that considers a general latent factor for entities of every entity type and context-dependent latent factors for every context in which the entities are involved. We learn a general latent factor for every entity and transfer matrices for every context to convert the general latent factors into a context-dependent latent factor. Experiments on two real life datasets from Epinions and Flixster demonstrate that HETEROMF substantially outperforms CMF, particularly for cold-start entities and for contexts where interactions in one contexts are dominated by other contexts.

Categories and Subject Descriptors

H.2.8.d [Information Technology and Systems]: Database Applications - Data Mining

General Terms

Algorithms, Measurement, Experimentation

Keywords

Recommendation, Matrix Factorization, Multi-Context, Cold Start

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media. WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil. ACM 978-1-4503-2035-1/13/05.

1. INTRODUCTION

With the rapidly growing amount of information available on the WWW, it becomes necessary to have tools to help users to select the relevant part of online information. To satisfy this need, recommender systems have emerged, e.g., there are popular recommenders for movies¹, books², music³, etc. Recommender are becoming a viable alternative and complement to search engines in helping users to find objects of interest. Typically, in a recommender system, a set of *users interact* with a set of *items*. The interactions vary, e.g., from users providing explicit scores for movies or books, to users listening to songs multiple times, to joining a community or favoriting a photo. We generically refer to all types of interactions in which users endorse items as *ratings*. The ratings expressed by users on items are stored in a rating matrix, which is typically extremely sparse [1, 10]. The classical task of a recommender is to predict the rating for user u on a non-rated item i based on the known ratings.

Matrix Factorization (MF) [13, 8] based models are the state-of-the-art methods for recommender systems. MF factors the rating matrix into two low rank matrices that represent latent factors for users and items. The model based on these factor matrices is then used for rating predictions. The typical setting in recommender systems is based on interactions between two types of entities. However, increasingly, there are social eco-systems where multiple entity types co-exist and interact with each other and across a variety of contexts, leading to multiple rating matrices. In other words, entities can have interactions in a Heterogeneous Information Network (HIN). Generally, in a heterogeneous network, entities from any two entity types can have interactions, with a weight (rating) indicating the level of endorsement. For example, in a social media service like LinkedIn⁴, there are entities of types user, company, university, etc., and there are interactions in the form of user-user, user-company, company-university, etc. The recommendation problem we study in this paper is *how to recommend entities of one type to those of another*, e.g., users to users (contacts), universities to companies (hiring), or professors to companies (expertise), leveraging the feedback data from the various interactions to predict the weight (rating) on interactions that have yet to happen. Recommendations in heterogeneous information networks can be valuable for applications like feed ranking. For instance, consider personalizing the news feeds in Facebook, Google+ or LinkedIn, where any entity can have news updates and entities from different types can get engaged in interactions, includ-

¹www.netflix.com

²www.amazon.com

³www.last.fm

⁴www.linkedin.com

ing the consumption of news feeds. This can make use of not just ranking based on content but also the predicted importance of the appropriate entity. E.g., if a student is more interested in certain companies, this can be factored in the way news feeds from various entities are ranked, for that student. Here, predicting the importance of one entity for another requires a recommendation model that can work on heterogeneous networks. Figure 1 illustrates the “schema” of a sample heterogeneous information network.

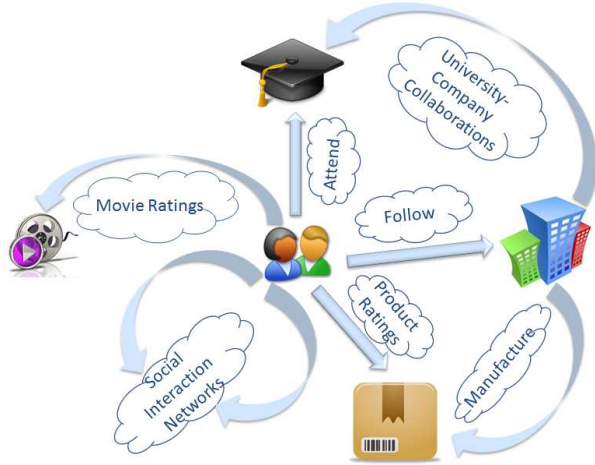


Figure 1: Schema of a sample Heterogeneous Information Network.

Collective Matrix Factorization (CMF) [17, 9] has been proposed to address the recommendation problem in heterogeneous networks. In CMF, every entity has a latent factor and interactions between two entities are governed by their corresponding latent factors. The latent factor of an entity is learnt based on the observed interaction data from all the contexts in which this entity is involved in interactions. However, a main issue with CMF is that *entities share the same latent factor across different contexts*. This is particularly problematic in two cases. First, suppose an entity e is cold-start in one context c . That is, e participates in very few interactions (ratings) in context c . Latent factors for entity e will be learnt mainly based on the interaction data from other contexts c' where e is not cold-start, and therefore the factors are not properly learned for the cold start entities in the context c . Second, suppose a context has significantly more interaction data compared to another context. Then the dominant context will dominate the learning process for the latent factors for entities shared in these two contexts. In this case, the latent factors for the shared entities are not learnt properly for the dominated context, even for non cold start entities in the dominated context.

Recall that we use rating as a general term to include the real valued item ratings in Epinions⁵ and Flixster⁶ and binary rating values such as joining a community in Facebook⁷ and LiveJournal⁸ or adding a photo to your favorite list in Flickr⁹. Also, visit counts on a Web page or the number of interactions among two entities are also indicators of ratings (weights) on these interactions. In other words, any type of user behavior demonstrating her evaluation (or

⁵www.epinions.com

⁶www.flixster.com

⁷www.facebook.com

⁸www.livejournal.com

⁹www.flickr.com

endorsement) of an entity can be regarded as a rating (binary or real values).

In this paper, we propose a context-dependent matrix factorization model, HETEROMF, that extends CMF and considers a general latent factor for every entity type and a context-dependent latent factor for every context in which the entity is involved. HETEROMF learns a general latent factor for every entity and transfer matrices for every context, to convert the general latent factors into context-dependent latent factors.

To test the effectiveness of our proposed method, we conduct a detailed set of experiments on two real life datasets from Epinions and Flixster. Our results demonstrate that HETEROMF substantially outperforms CMF, particularly for cold-start entities and for contexts where the size of the interaction data is dominated by other contexts.

There has been a large body of recent work on link prediction in heterogeneous information networks [21, 19]. We review these works in detail in Section 3, but note here that their main focus is analyzing how topological structures in heterogeneous networks affect link (relationship) formation and using this to predict whether (and, in some cases, when) an entity will form a relationship with another. As such our goal and methodology are different from theirs. The main goal of this paper is to learn the latent factors governing the behavior of entities in different contexts (rather than using topological structures), and address the issues mentioned for the state-of-the-art CMF methods.

In this paper, we make the following contributions:

- We introduce the novel HETEROMF model, a generalized CMF based model that considers general latent factors for entities as well as context specific latent factors for the contexts in which entities are involved in interactions.
- We use a transfer matrix to convert base latent factors into context specific latent factors. Transfer matrices are learnt per context.
- HETEROMF is particularly helpful for cold start entities, and for the contexts that are dominated by the interactions from other contexts.
- We perform experiments on two *real life* data sets from Epinions and Flixster. Experimental results show that HETEROMF achieves substantial RMSE gain compared to CMF, particularly for cold start entities, and for the contexts dominated by other contexts.

The rest of the paper is organized as follows: Preliminaries and problem definition are discussed in Section 2, followed by some of more recent related works in Section 3. Our proposed HETEROMF model is introduced in Section 4. Also we discuss the inference algorithm for HETEROMF in Section 4. In Section 5, we present our experimental results on real life datasets. Finally, we conclude the paper and discuss some future work in Section 6.

2. PRELIMINARIES

In this section, we review some preliminaries and formally state the problem studied in this paper.

A *heterogeneous information network* (HIN) G consists of N entity types denoted by $\mathcal{U} = \{\mathbb{U}_1, \dots, \mathbb{U}_N\}$ and τ contexts denoted by $\mathcal{L} = \{L_1, \dots, L_\tau\}$. Each context L_l consists of interactions among entities from two *participating* entity types \mathbb{U}_n and \mathbb{U}_m , where $\mathbb{U}_n, \mathbb{U}_m \in \mathcal{U}$. For example *movie ratings* is an instance of a context in which entities of type *user* interact with entities of

type *movie* and the interactions result in interaction weights that we generically call *ratings*. Note that the two participating entity types of a context can be the same: e.g., in the context of social networks both entity types are user. Similarly, more than one context can bring about interactions between the same pair of entity types: e.g., a social network (with explicit links) as well as a co-authorship network (based on common published papers) are two contexts which relate users with users.

The observed interactions in a context L_l are stored in a context-specific rating matrix R_l . R_l is typically a very sparse matrix. Every cell in this matrix, $r_{u,v}^l$, denotes the rating (interaction weight) observed from entity $u \in \mathbb{U}_n$ on entity $v \in \mathbb{U}_m$. Here \mathbb{U}_n and \mathbb{U}_m are the entity types participating in the context L_l . The set of all rating matrices corresponding to the various contexts is denoted \mathcal{R} .

We next formally state the problem we study in this paper.

DEFINITION 1 (PROBLEM STUDIED). Given a heterogeneous information network $G = (\mathcal{U}, \mathcal{L}, \mathcal{R})$, a context $L_l \in \mathcal{L}$ consisting of interactions among $\mathbb{U}_n \in \mathcal{U}$ and $\mathbb{U}_m \in \mathcal{U}$, and entities $u \in \mathbb{U}_n$ and $v \in \mathbb{U}_m$ such that $r_{u,v}^l$ is unknown, predict $r_{u,v}^l$ using the observed ratings in \mathcal{R} . \square

Matrix Factorization (MF) is one of the common techniques for model-based recommendation. MF has been proposed in order to perform predictions for a single user-item rating matrix, for a single context, unlike the setting in this paper. In MF, each user and each item is associated with a K dimensional latent factor vector [6]: the latent factor of user u is denoted U_u and stored as the u^{th} row of user factor matrix U . The latent factor of item i is denoted V_i and stored as the i^{th} row of item factor matrix V .

In order to learn the latent factors of users and items, [13] employs probabilistic matrix factorization to factor the user-item matrix into the product of user and item latent factors. The conditional probability of the observed ratings is defined as:

$$p(R|U, V, \sigma_R^2) = \prod_{u=1}^N \prod_{i=1}^M \left[\mathcal{N}(R_{u,i} | U_u^T V_i, \sigma_r^2) \right]^{I_{u,i}^R} \quad (1)$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2 , and $I_{u,i}^R$ is the indicator function that is equal to 1 if u has rated i and equal to 0 otherwise. Also, zero mean Gaussian priors are assumed for user and item factors:

$$p(U|\sigma_U^2) = \prod_{u=1}^N \mathcal{N}(U_u | 0, \sigma_U^2 \mathbf{I}), \quad p(V|\sigma_V^2) = \prod_{i=1}^M \mathcal{N}(V_i | 0, \sigma_V^2 \mathbf{I}) \quad (2)$$

Figure 2 presents the graphical model representing matrix factorization. The latent factors of users and items are learnt in MF using the observed rating matrix. To predict the rating of user u on item j in the test phase, the inner product of learnt latent factors U_u and V_j is computed as the predicted rating.

The matrix factorization based method, essentially considered the state-of-the-art method for recommendation, has been recently extended to incorporate other factors in the factorization of the rating matrix: these include modeling at multiple scales [3], incorporating users with similar rating patterns [6], and taking the timestamps into account [7]. The last approach [7] won the Netflix grand \$1M prize¹⁰. Matrix factorization is the basis for building extended models such as collective matrix factorization, and our proposed model, HETEROMF.

¹⁰<http://www.netflixprize.com/>

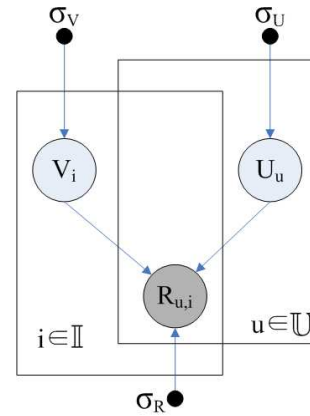


Figure 2: Graphical model representing MF.

3. RELATED WORK

In this section, we review some related work on matrix factorization, collective matrix factorization and recommendation and link prediction in heterogeneous information networks.

As discussed in the previous section, matrix factorization is proposed for single context domains, where there is only one rating matrix. However, users can be involved in interactions across multiple contexts. In this case, one could fit the rating matrix in each context separately. However, this approach would not take advantage of any correlations between contexts. In other words, information from one context does not propagate to another context. This approach is called separate matrix factorization (SMF) and we use SMF as one of our baseline models in our experiments.

Singh and Gordon [17] proposed Collective Matrix Factorization (CMF) for learning in multi-context domains. CMF decomposes the rating matrix in every context into a product of latent factor matrices of the entities from the two participating entity types of the context. Whenever an entity type participates in more than one context, the latent factors for that entity type are shared among the contexts in which this entity participates. Figure 3 shows an example graphical model representing a CMF model. In this model, there are two contexts (with observed ratings R and O) and three entity types (U, V, Z). E.g., it could represent users (U) rating movies (V) and books (Z). The graphical model representing CMF for more than two contexts is simply a generalization of the one shown in Figure 3.

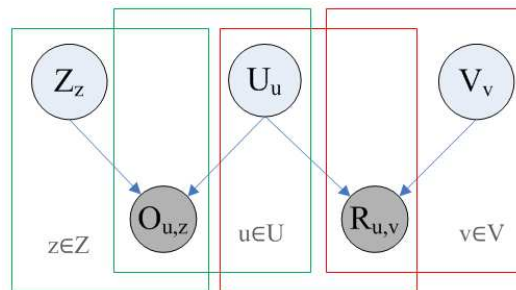


Figure 3: Graphical model representing a simple Collective Matrix Factorization (CMF), with only two contexts.

In CMF, the latent factor of an entity from a specific entity type is learned based on the observed ratings from the contexts in which this entity type participates. However, a main issue with CMF is that entities share the same latent factor across different contexts. We conjecture that this is particularly problematic in two cases. First, latent factors for entities that are cold-start in a context will be learned mainly based on the data from other contexts where it is not cold-start, and therefore the factors are not properly learned for the cold-start context. In practice, cold start entities are entities that have very few (normally less than 5) observed ratings that they participate in.

Second, if a context has more interaction data compared to another context, then the dominant context will dominate the learning process for the latent factors for entities shared in these two contexts. In other words, the latent factors for all entities in the shared entity type are learnt mainly based on the dominating context and the dominated context has very little effect on the learned latent factors. Therefore, the latent factors are not properly learned for representing the dominated context. In this paper, we address these issues by introducing context-dependent latent factors besides having general factors for entities in different entity types. CMF is our main competitor in the experiments and we show that our proposed model addresses the above issues with CMF and substantially outperforms it.

Collective matrix factorization has also been used in [9]. Lippert et al. [9] consider user attributes (e.g., gender, age, etc.) as different contexts. However, in recent papers [24, 2, 4] user attributes (features) are considered as the prior for user latent factors, and a transfer matrix [24] is used to generate a latent factor using user attributes. In our research, we do not assume we have access to entity attributes. However, we use the same idea as in [24, 2, 4] to generate context-specific latent factor using the general latent factor for the entity and a context-dependent transfer matrix.

Recently, Yang et al. [22] employed collective matrix factorization for recommendation in social rating networks where there is a rating matrix between users and items and a social network among users. The proposed model in this paper considers the social network and user ratings as two contexts and decomposes each context into the product of latent factors of the entities corresponding to the participating entity types of that context. The model proposed in [22] can be represented by the graphical model demonstrated in Figure 3 by setting $Z = U$ and letting O denote social relations. Note that the social network in this paper is considered to be undirected. Among other things, the authors showed that their approach leads to better prediction accuracy than using the ratings alone or using the social network alone.

There is an important body of research on link prediction in heterogeneous information networks [21, 23, 19, 14, 18]. These works particularly focus on how topological structures in heterogeneous networks affect the relationship building and whether an entity engages in an interaction with another entity or not. Yu et al. [23] and Sun et al. [18] used the concept of “meta-path”, first introduced in [20], to define features for a potential interaction between entities a and b , and further used these features to learn a regression function to compute the probability of a link (interaction). In the meta-path based approach, a set of path patterns from entity a to b is selected and the number of such paths is computed to determine the path-specific features. Computation of paths between two entities is performed on the fly (memory-based) which requires exploring the HIN and makes the meta-path based methods slower in prediction compared to model-based approaches such as CMF that do not require to access the raw data after the learning phase. Meta-path based methods also have a training phase to learn the

regression weights to combine the various topological features to compute the probability of an interaction.

Shi et al. [16, 15] proposed a model to use the labeled data from other related (heterogeneous) sources to help in predicting the target task even if they have different feature spaces and distributions. This paper proposes a solution and discusses the conditions where this is possible and highly likely to produce better results. This model uses spectral embedding to unify the different feature spaces of the target and source data sets, even when they have completely different feature spaces. The principle is to cast into an optimization objective that preserves the original structure of the data, while at the same time, maximizes the similarity between the two [16]. Next, a sample selection strategy is applied to select only those related source examples. Finally, a Bayesian-based approach is applied to model the relationship between different output spaces. While these works are very interesting, they mainly deal with transferring feature spaces from a domain to another domain, rather than learning from interactions in other contexts which is our main focus.

4. PROPOSED HETEROMF

In this section we introduce our proposed model, HETEROMF, that extends collective matrix factorization and incorporates context dependent latent factors for entities, in their interactions in different contexts. As discussed before, the main problem with CMF models is that the latent factors for entities of an entity type are shared among all the contexts that this entity type participates in. Recall that this is particularly problematic for cold-start entities, and for contexts dominated by other contexts.

To address these issues, we propose HETEROMF, a generalized context dependent collective matrix factorization based approach. In our proposed model, each entity of every entity type has a base latent factor. For each context in which this entity type participates, entities of that type have a context specific latent factor. To elaborate on the intuition, suppose that in a domain, users can rate movies and books. In this example, users have some general factors representing their general rating behavior. However, the factors governing users’ behavior in the context of movies is different from her factors in the context of books. Context specific factors depend on the base factors and can be derived from the base factors of an entity. In HETEROMF, a transfer matrix is used to transfer the base factors to the context specific factors in each context. The following is the list of latent factors used in HETEROMF:

- Latent factors for entities of type \mathbb{U}_n are denoted by U_n , $n \in [1, N]$. The latent factor for an entity $u \in \mathbb{U}_n$ is denoted by $U_{n,u}$.
- Entities of an entity type have their own context specific latent factors in the contexts L_l this entity type participates in and the context specific latent factors are denoted by U_n^l . In HETEROMF, context specific latent factors U_n^l are generated using the base latent factors U_n and a context specific transfer matrix M_n^l . The context specific latent factor for an entity $u \in \mathbb{U}_n$ and for context L_l is denoted by $U_{n,u}^l$.
- Every context L_l has two entity types. Therefore every context L_l is associated with two context-dependent entity latent factors U_n^l and U_m^l . In some cases such as social relations, we can have $n = m$. Note that for each context, there are two separate transfer matrices M_n^l and M_m^l , unless the context is an undirected network ($n = m$). It should be noted that if a network is undirected, then $M_n^l = M_m^l$.

In our proposed HETEROMF model, the observed ratings for a context L_l are generated using the context specific latent factors for the entities whose entity types participate in that context, as follows:

$$R_{u,v}^l \sim \mathcal{N}(U_{n,u}^l U_{m,v}^l, \sigma_{r,l}^2) \quad (3)$$

In this equation and in what follows, we use M' to denote the transpose of matrix M . Context specific latent factors U_n^l are generated using the product of the base latent factors U_n and a context specific transfer matrix M_n^l . If $U_{n,u}^l$ is a k_1 -dimensional column vector and $U_{n,u}$ is a k_2 -dimensional column vector, then M_n^l is a $k_2 \times k_1$ matrix that converts the base latent factor to a context specific latent factor. M_n^l should be learned in the training phase.

$$U_{n,u}^l \sim \mathcal{N}(M_n^l U_{n,u}, \sigma_{l,n}^2 \mathbf{I}) \quad (4)$$

In the above Equation, \mathbf{I} is the identity matrix. Similar to regular MF, every base latent factor has a zero-mean normal prior. Note that, if the attributes for different entity types (e.g., age and gender for users or location for companies) were available, we could use them as the mean of the normal distribution for prior, instead of a zero mean normal prior.

$$U_{n,u} \sim \mathcal{N}(\vec{\mathbf{0}}, \sigma_n^2 \mathbf{I}) \quad (5)$$

Figure 4 demonstrates the graphical model representing the HETEROMF model. In this graphical model, σ_n denotes the variance for the distribution of base latent factors U_n , $\sigma_{l,n}$ is the variance for context specific latent factors U_n^l , and $\sigma_{r,l}$ is the variance for the observed ratings in the context L_l .

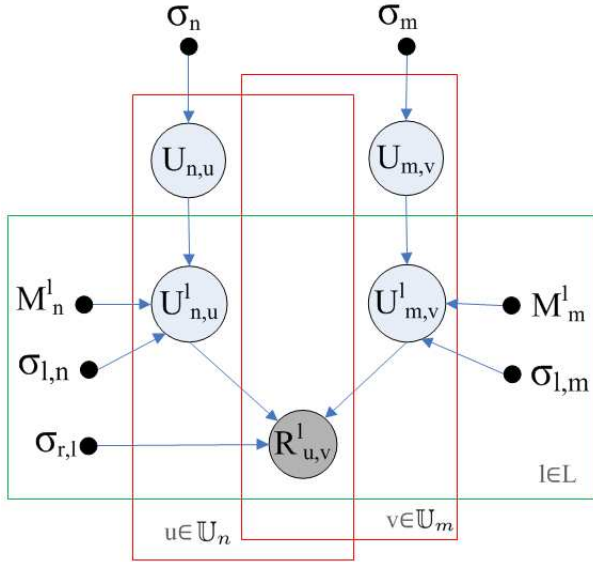


Figure 4: The graphical model representing HETEROMF.

HETEROMF is particularly helpful for entities that are cold-start in a context but are active in other contexts. HETEROMF uses the base factors of those entities and transfers them to the context specific latent factors using the transfer matrix for that context, without being dominated by the latent factors from the contexts in which these entities are not cold-start. Also, even if a context is dominated by another context, the context specific latent factors will not

be dominated. Although the base factors for the shared entity type can be dominated, notice that the transfer matrices are learned per context, leading to a proper learning of the context specific latent factors and resolving the shared latent factors' issues in CMF.

Using the graphical model demonstrated in Figure 4, the complete data likelihood (joint likelihood) in HETEROMF is computed as follows:

$$P(R, \theta | \eta) = \prod_l P(R^l | U_n^l, U_m^l, \sigma_{r,l}) \times \prod_l (P(U_n^l | U_n, M_n^l, \sigma_{l,n}) \times P(U_m^l | U_m, M_m^l, \sigma_{l,m})) \times \prod_n P(U_n | \sigma_n) \quad (6)$$

In the above equations, we have:

$$\eta = \{\cup_n \sigma_n, \cup_l \sigma_{l,n}, \cup_l \sigma_{l,m}, \cup_l \sigma_{r,l}, M = \cup_l \{M_n^l, M_m^l\}\}$$

$$\theta = \{\cup_l U_n^l, \cup_l U_m^l, \cup_n U_n\}, R = \cup_l \{R^l\}$$

It should be noted that if R^l is an undirected network, then $M_n^l = M_m^l$ and therefore $U_{n,u}^l = U_{m,u}^l$. Using Equations (3)-(6), the log-likelihood of the complete data is computed as follows:

$$\begin{aligned} \mathcal{L} = & -\frac{1}{2} \sum_l \sum_{u \in U_n^l} \sum_{v \in U_m^l} \left(\frac{1}{\sigma_{r,l}^2} (r_{u,v}^l - U_{n,u}^l U_{m,v}^l)^2 + \log \sigma_{r,l}^2 \right)^{I_{u,v}^{R,l}} \\ & - \frac{1}{2} \sum_l \sum_{u \in U_n} \left(\frac{1}{\sigma_{l,n}^2} (U_{n,u}^l - M_n^l U_{n,u})^2 + k^l \log \sigma_{l,n}^2 \right) \\ & - \frac{1}{2} \sum_l \sum_{u \in U_m} \left(\frac{1}{\sigma_{l,m}^2} (U_{m,u}^l - M_m^l U_{m,u})^2 + k^l \log \sigma_{l,m}^2 \right) \\ & - \frac{1}{2} \sum_n \sum_{u \in U_n} \left(\frac{1}{\sigma_n^2} (U_{n,u})^2 + k_n \log \sigma_n^2 \right) \quad (7) \end{aligned}$$

In the above equations, $I_{u,v}^{R,l}$ is an indicator that returns 1 if entity u has a relation with entity v in the context L_l , and returns 0 otherwise. Also, k^l is the dimensionality of latent factors in context L_l , and k_n is the dimensionality of factors for entity type U_n .

4.1 Inference in HeteroMF

We use expectation maximization (EM) to fit HETEROMF and learn the latent factors and model parameters. In the E step, we compute the expected log-likelihood with respect to the latent variables ($E_{\theta|R,\eta}(\mathcal{L})$). In the M step, we maximize the expected likelihood with respect to the model parameters (η) to compute the model parameters. The EM iterations are repeated until convergence.

The expected log-likelihood is computed as follows:

$$\begin{aligned}
E_{\theta|R,\eta}(\mathcal{L}) = & \\
& -\frac{1}{2} \sum_l \sum_{u \in U_n^l} \sum_{v \in U_m^l} \left(\log \sigma_{r,l}^2 + \frac{(r_{u,v}^l - \widehat{S}_{u,v}^l)^2 + \text{Var}[S_{u,v}^l]}{\sigma_{r,l}^2} \right)^{I_{u,v}^{R,l}} \\
& -\frac{1}{2} \sum_l \sum_{u \in U_n} \left(k^l \log \sigma_{l,n}^2 + \right. \\
& \quad \left. \frac{\sum_t \left(E_{\theta|R,\eta}((U_{n,u}^l[t] - M_n^l[t,]U_{n,u})^2) \right)}{\sigma_{l,n}^2} \right) \\
& -\frac{1}{2} \sum_l \sum_{u \in U_m} \left(k^l \log \sigma_{l,m}^2 + \right. \\
& \quad \left. \frac{\sum_t \left(E_{\theta|R,\eta}((U_{m,u}^l[t] - M_m^l[t,]U_{m,u})^2) \right)}{\sigma_{l,m}^2} \right) \\
& -\frac{1}{2} \sum_n \sum_{u \in U_n} \left(k_n \log \sigma_n^2 + \frac{\sum_t (\widehat{U}_{n,u}[t]^2 + \text{Var}[U_{n,u}[t]])}{\sigma_n^2} \right) \quad (8)
\end{aligned}$$

In the above equation, $S_{u,v}^l = U_{n,u}^l U_{m,v}^l$, $\widehat{S}_{u,v}^l$ is the sample mean of $S_{u,v}^l$, $\text{Var}[S_{u,v}^l]$ is the sample variance of $S_{u,v}^l$. Also $U_{n,u}[t]$ is the t^{th} element in the vector representing $U_{n,u}$ ¹¹.

4.1.1 E Step

We use Gibbs sampling in the E step. We draw T samples for all the latent variables and compute the sufficient statistics (Monte Carlo means and variances of the latent variables) to be used in the M step.

In the E step, we sample the latent variables from their distributions given the rest of random variables. For example, to sample $U_{n,u}$, we sample from $P(U_{n,u}|rest)$. This probability is embedded in the complete data likelihood and since all distributions are normal, $P(U_{n,u}|rest)$ is also normal. Generally, if $f(\mathbf{x})$ is a multivariate normal density function with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$. Then, we have:

$$\boldsymbol{\mu} = \boldsymbol{\Sigma} \left(\frac{d}{dx} \log f(\mathbf{x}) \right) \Big|_{\mathbf{x}=\mathbf{0}} \text{ and } \boldsymbol{\Sigma} = - \left(\frac{d^2}{dx^2} \log f(\mathbf{x}) \right)^{-1}$$

Now, for latent variables U_n and U_n^l we use the above lemma to compute the mean and variance of the normal density functions to sample these latent variables from:

$$\text{Var}[U_{n,u}|rest] = \left(\frac{1}{\sigma_n^2} \mathbf{I} + \sum_{l \in C_n} \frac{M_n^l M_n^l}{\sigma_{l,n}^2} \right)^{-1} \quad (9)$$

$$E[U_{n,u}|rest] = \text{Var}[U_{n,u}|rest] \left(\sum_{l \in C_n} \frac{M_n^l U_{n,u}^l}{\sigma_{l,n}^2} \right) \quad (10)$$

In the above equations, C_n is the set of contexts in which entity type U_n is involved. Similarly, the mean and the covariance matrix for $U_{n,u}^l$ is computed as follows:

¹¹Note that to improve readability in Eq. (8), we used $E_{\theta|R,\eta}((U_{n,u}^l[t] - M_n^l[t,]U_{n,u})^2) = (U_{n,u}^l[t] - M_n^l[t,]U_{n,u})^2 + \text{Var}[U_{n,u}^l[t]] + M_n^l[t,] \text{Var}[U_{n,u}] M_n^l[t,] - 2M_n^l[t,] \text{Cov}[U_{n,u}^l[t], U_{n,u}]$.

$$\text{Var}[U_{n,u}^l|rest] = \left(\frac{1}{\sigma_{l,n}^2} \mathbf{I} + \sum_{v \in U_m^l} \frac{U_{m,v}^l U_{m,v}^l}{\sigma_{r,l}^2} \right)^{-1} \quad (11)$$

$$E[U_{n,u}^l|rest] = \text{Var}[U_{n,u}^l|rest] \left(\sum_{v \in U_m^l} \frac{r_{u,v}^l U_{m,v}^l}{\sigma_{r,l}^2} + \frac{M_n^l U_{n,u}}{\sigma_{l,n}^2} \right) \quad (12)$$

In the above equations, $U_u^{R,l}$ denotes the entities (from U_m^l) that have relation with u in the context l .

4.1.2 M Step

In the M step, we maximize the expected likelihood, $E_{\theta|R,\eta}(\mathcal{L})$, with respect to model parameters ($\boldsymbol{\eta}$) to compute the updated values for the model parameters. To do so, we set the derivative of $E_{\theta|R,\eta}(\mathcal{L})$ with respect to every model parameter to zero and compute the model parameters as follows:

$$\sigma_{r,l}^2 = \frac{\sum_{u \in U_n^l} \sum_{v \in U_m^l} \left((r_{u,v}^l - \widehat{S}_{u,v}^l)^2 + \text{Var}[S_{u,v}^l] \right)^{I_{u,v}^{R,l}}}{|R^l|} \quad (13)$$

$$\sigma_n^2 = \frac{\sum_{u \in U_n} \sum_t (\widehat{U}_{n,u}[t]^2 + \text{Var}[U_{n,u}[t]])}{k_n |U_n|} \quad (14)$$

$$\sigma_{l,n}^2 = \frac{\sum_{u \in U_n} \sum_t \left(E_{\theta|R,\eta}((U_{n,u}^l[t] - M_n^l[t,]U_{n,u})^2) \right)}{k^l |U_n|} \quad (15)$$

$$\begin{aligned}
M_n^l[t,] = & \left(\sum_{u \in U_n} (\widehat{U}_{n,u}^l[t] \widehat{U}_{n,u} + \text{Cov}[U_{n,u}^l[t], U_{n,u}]) \right)' \\
& \times \left(\sum_{u \in U_n} (\widehat{U}_{n,u} \widehat{U}_{n,u}' + \text{Var}[U_{n,u}]) \right)^{-1} \quad (16)
\end{aligned}$$

Now, we continue to the next EM iteration using the updated model parameters. We continue performing E and M steps until convergence of the learned parameters.

5. EXPERIMENTS

In this section, we first describe the datasets used in our experiments and our experimental setting and design, and then present the experimental results.

5.1 Datasets

One of the bottlenecks to research in heterogeneous network analysis has been the lack of publicly available datasets featuring a large number of entity types, numerous contexts, and rich interaction data. Many previous papers on heterogeneous information networks have tended to use bibliographic datasets [19, 23]. These datasets contain the entity types, authors, institutions, publications, and venues, to name a few. However, there are no explicit ratings in these datasets, and the contexts in bibliographic networks are not real context where entities interact. Ideal datasets for this research, particularly for recommendations research, would be the real heterogeneous networks from online social media services such as

Facebook, Google+, or LinkedIn where there are multiple entity types such as users, universities, companies, etc. and these entities interact in several contexts. However, due to lack of public data available for such heterogeneous networks featuring many entity types and multiple contexts, we use two publicly available data sets from Epinions.com and Flixster.com. These two datasets are in fact social rating networks [5] and each one consists of two contexts: a social network among users, and an item rating matrix. In this subsection we briefly introduce these two datasets.

Epinions.com is a product reviewing website in which users can express trust on other users besides writing reviews on various products. We used the version of the Epinions dataset¹² published by Richardson and Domingos [12]. The social relations in Epinions are in fact trust relations, which are directed. Flixster.com is a social networking service in which users can rate movies and create a social network. The social relations in Flixster are friendship relations and undirected. The Flixster data set has been crawled by Jamali and Ester [5] and has been made publicly available for the research community¹³. Ratings in Flixster and Epinions are real values in the range [1,5]. The social relations in both data sets are binary, indicating whether or not a social relation exists. General statistics of the Flixster dataset and the Epinions data set are shown in Table 1. Between these two datasets, we capture both directed and undirected social networks and two contexts, one involving binary ratings and the other real ratings.

Table 1: General statistics of the Flixster and Epinions

Statistics	Flixster	Epinions
Users	1M	71K
Social Relations	26.7M	508K
Ratings	8.2M	575K
Items	49K	104K

5.2 Experimental Setups

We implemented the algorithms in R and in C: R scripts are used to handle the input matrices and high level routines, and C is mainly used for more complex processing. Our experiments were run on a single core of a multi-core machine with an Intel Xeon CPU X5570 2.93GHz and 8M cache. While the machine had 64GB memory, our implementation of HETEROMF consumed only 1GB memory on the largest dataset (Flixster).

We perform 5-fold cross validation in our experiments. In each fold we have 80% of data as the training set and the remaining 20% as the test set.

The evaluation metric we use in our experiments is RMSE, defined as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,v) \in R_{test}} (r_{u,v} - \hat{r}_{u,v})^2}{|R_{test}|}} \quad (17)$$

where R_{test} is the set of all entity pairs (u, v) in the test data. Note that RMSE values for different contexts are computed separately and R_{test} in the above equation is the test data for a specific context. We report the results for different contexts in both Epinions and Flixster separately.

To evaluate the performance of our proposed model, we consider three comparison partners, i.e., competing approaches:

- HeteroMF: The context-specific factorization based model proposed in this paper.
- CMF: The collective matrix factorization method [9].
- SMF: Separate matrix factorization, where different contexts are learned separately using separate latent factors. No information is propagated across contexts.

In the datasets used in our experiments, we only have entity types of *user*, *product*, and *movie*. Entity type *user* is the only entity type that is shared among two contexts. To perform a comprehensive set of experiments and evaluate the strength of our model in all the aspects and issues that we identified for CMF, we design and perform different sets of experiments. We report experimental results on three kinds of users:

- *AllUsers*: All the users in the system.
- *Cold Start*: Cold-start users for a specific context who are users that have rated less than 5 (but more than zero) items in one context, and 5 or more ratings in the other context.
- *Inactive*: Users who have no ratings in a context, but have 5 or more ratings in the other context. These users are particularly important for the case that the latent factors of a user are properly learned in a context and now this user wants to start expressing ratings in a new context. Transferring the learnt factors across contexts to model the rating behavior for these users is very important.

The three kinds of users above cover key settings which we believe should both highlight the advantage of our proposed model and expose its limitations, if any. In both datasets, as such none of the ratings or the social relations dominate each other. To overcome this, we simulate a dataset where one context dominates another one, by selecting a subset of the ratings in each dataset to represent a context. In Epinions, if we only consider the category of “online-store”, then the ratings (22K) will be dominated by the social relations (508K). Similarly in Flixster, if we only consider the movies from the “Horror” genre, then the ratings (245K) will be dominated by the social relations (26M). In our experimental results we refer to these data sets as *Epinions-dominant* and *Flixster-dominant*.

5.3 Experimental Results

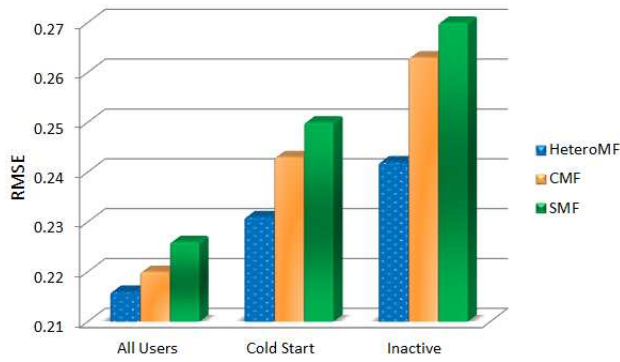
In this subsection, we report our experimental results on the different sets of experiments introduced in the previous subsection for both Flixster and Epinions.

Epinions.

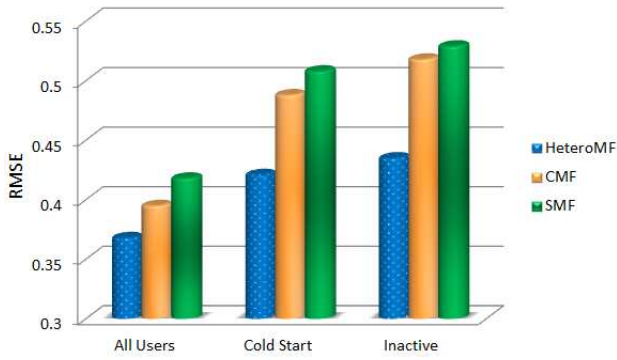
Figure 5 compares the RMSE of the comparison partners for different sets of users in both the contexts – rating and trust, for Epinions. In both contexts, HeteroMF outperforms the comparison partners. Notice that HeteroMF’s improvement over CMF is substantially more than the latter’s improvement over SMF. Figure 6 shows RMSE improvement of HeteroMF over CMF for various kinds of users. It can be easily seen that HeteroMF achieves a substantial RMSE improvement over CMF for cold start users and inactive users, compared with all users. In the rating context, the RMSE gain for cold-start users (4.9%) is more than 2.5 times the RMSE gain for all users, and the RMSE gain for inactive users is more than 1.6 times the RMSE gain for cold-start users. Similar behavior is exhibited in the trust context. The absolute value of the RMSE gain of HeteroMF for each kind of users is more than for the rating context. But the increase in the RMSE gain when we go from all users to cold-start users to inactive users is more modest compared to the rating context.

¹²<http://alchemy.cs.washington.edu/data/epinions/>

¹³www.sfu.ca/~sja25/datasets/



(a) Rating



(b) Trust

Figure 5: RMSE results for comparison partners in different contexts for Epinions.

Flixster.

Figure 7 shows the results for the Flixster dataset. As shown in this figure HeteroMF outperforms all the comparison partners for all three sets of users in both contexts. Unlike Epinions, HeteroMF’s RMSE gain over CMF is comparable to the latter’s RMSE gain over SMF. Figure 8 depicts how the RMSE gain of HeteroMF over CMF varies depending on the kind of users. It clearly shows that similar to the case for Epinions, HeteroMF achieves substantial RMSE gain for cold start users and inactive users, compared with all users.

Analysis.

Experimental results on cold start users and inactive users show that CMF can not properly learn the latent factors for entities that have not expressed enough ratings in one context, but have many ratings in another context. This is mainly due to the fact that the latent factors for users are shared across different contexts in CMF. For cold start users in a context, their latent factors are dominated by the other context. On the other hand, HeteroMF learns general latent factors for users and context specific latent factors in rating and social contexts. Transfer matrices are used to convert a general latent factor to a context specific factor.

In other words, the main advantage of HETEROMF over CMF is that HETEROMF decomposes the latent factors of an entity into a general latent factor and a context dependent transfer matrix. General factors are learnt based on the observations in all contexts. However, the transfer matrix is learnt for a specific context only. In HETEROMF, even if a user has no ratings it does not matter because his general factors are learnt based on other contexts and



Figure 6: Comparison of RMSE improvement for cold start users, all users, and inactive users in Epinions.

to tailor this factor to a context dependent one HETEROMF uses a transfer matrix that is properly tuned for this specific context and is in charge of translating general factors into context specific factors.

Dominated Contexts.

Figure 9 presents the RMSE for the dominated context (rating) in *Epinions-dominant*. As shown in this figure and in Figure 10, HeteroMF achieves a substantial RMSE improvement over CMF even for all users in Epinions-dominant on the dominated context, namely rating. HeteroMF outperforms CMF by 11% in the rating context which is very impressive, compared to less than 2% for the Epinions’ complete dataset. Similar results are achieved for *Flixster-dominant* (see Figures 11 and 10).

CMF vs. SMF.

There are some interesting and somewhat surprising results shown in Figures 9 and 11 in addition to those discussed earlier. In both datasets, SMF outperforms CMF in the rating context! This shows that when a context dominates another context, it is beneficial for the dominated context to be learned separately from the dominating context. Indeed, by learning the dominated context along with the dominating context, CMF suffers on the dominated context, compared with SMF. So far, all the experiments we have discussed convincingly highlight the weaknesses of CMF exactly as we outlined in the earlier sections and precisely under the settings discussed, w.r.t. different kinds of users and different contexts.

What is a context?.

It should be noted that there are different genres in Flixster and different product categories in Epinions which can potentially be considered as different contexts. Syntactically, we can of course regard them as contexts. Can we expect the same kind of behavior from these “contexts” that we observed for the rating vs. social/trust contexts in our experiments above? To test this, we performed experiments on these settings and the results were not striking, in that the RMSE gain of HeteroMF over CMF was not substantial (Even for some genres/categories, CMF performed slightly better.). We believe that this is mainly due to the fact that although different genres can potentially represent different contexts, in reality users’ behavior toward them tends to be very similar. Therefore, direct propagation of information from one context to another one (as is done by CMF) is not going to harm cold-start users. In other words, if a user is cold start in one genre but active in another genre, he/she is actually not a cold-start user and his/her ratings in the active context happen to faithfully capture his behavior in the

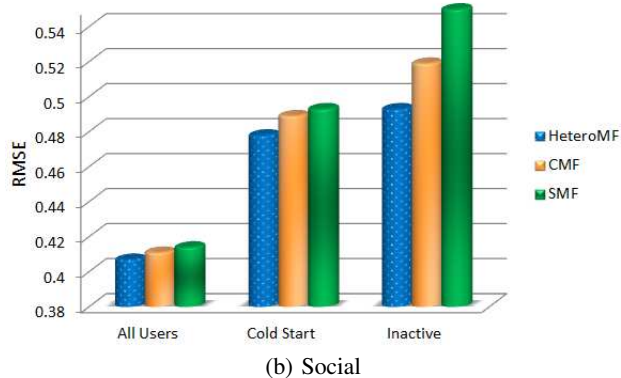
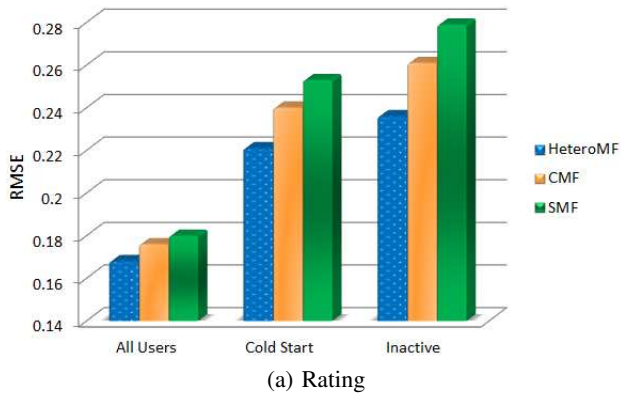


Figure 7: RMSE results for comparison partners in different contexts in Flixster.

so-called cold-start context, with high accuracy. E.g., there is not a great difference in the rating behavior of “Thriller” movies compared to “Comedy” movies. However, social relations and ratings are fairly independent and are genuinely separate contexts. Indeed, we can see the effect of multiple contexts much better in this setting, as shown in our experiments. Similar remark hold for the Epinions dataset.

5.3.1 Running Time

In our experiments, CMF converged in about 50 iterations, while HETEROMF took about 100 iteration to converge. Table 2 compares the actual average running time for an EM iteration in both HETEROMF and CMF on all datasets. As shown in this table, an EM iteration in CMF, on an average, runs more than 4 times faster than a corresponding EM iteration in HETEROMF, for all datasets used in our experiments. Slower iterations in HETEROMF are mainly due to the fact that HETEROMF considers context-specific factors. Also, to facilitate the propagation across contexts, in the E step we first sample context-specific factors, then the base factors, and then we do one more round of sampling for context-specific factors to propagate the information from the observed data to the base factors and from the base factors back to the contexts.

Note that although CMF is faster than HETEROMF in the learning phase, both methods are equally fast in the prediction (test) phase, when they simply use the learned latent factors. In real industrial strength applications, substantial accuracy improvements for cold-start entities and dominated contexts, that are achieved by HETEROMF compared to CMF, tend to be very important for new user engagement and for controlling information propagation from



Figure 8: Comparison of RMSE improvement for cold start users, all entities, and inactive users in Flixster.

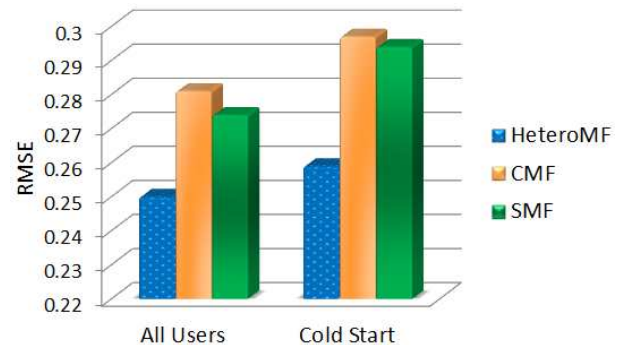


Figure 9: RMSE results for comparison partner in the dominated context (rating) for Epinions-dominant, where a context (rating) is dominated by another one (trust).

large contexts to much smaller sized contexts. Thus, the increased learning time is a price that may be worth paying in practice. It is worth noting, however, that the learning phases in both HETEROMF and CMF are highly parallelizable, which may allow us to save on learning time significantly.

6. CONCLUSIONS

Traditional recommender systems focus on interactions between users and items in a single context and use past interactions, expressed as ratings, to make predictions about future ratings users may give items they have not interacted with before. Methods based on matrix factorization are the state of the art for making these predictions. However, in heterogeneous information networks, entities of different types interact with each other, often across a variety of contexts. This leads to multiple rating matrices. Collective matrix factorization is the state of the art for rating predictions in such settings. We identified certain fundamental issues with the way in which CMF models and approaches the rating prediction problem in this setting. We proposed a context-dependent matrix factorization model, HETEROMF, that considers a general latent factor for entities of every entity type and context-dependent latent factors for every context in which the entities participate. HETEROMF learns a general latent factor for every entity and trans-

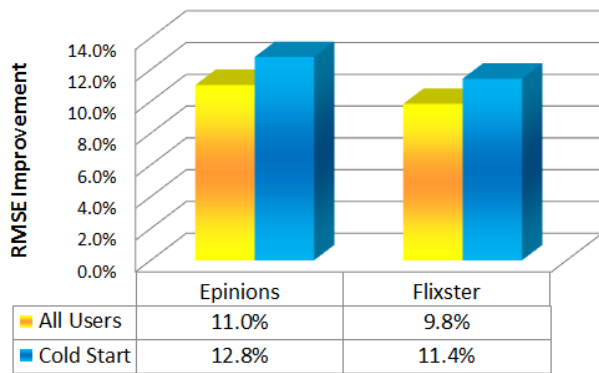


Figure 10: Comparison of RMSE improvement for cold start users and all users in the dominated context (ratings). The results are for both Epinions and Flixster for the case that a context (rating) is dominated by another one (social or trust).

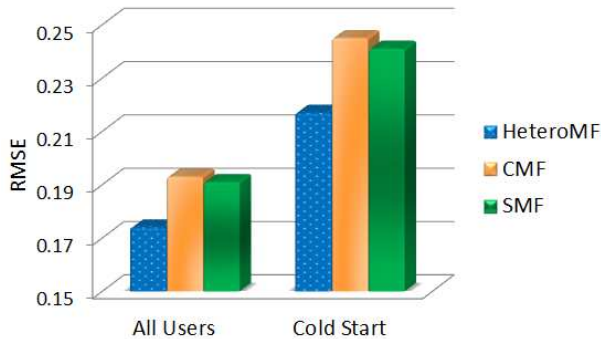


Figure 11: RMSE results for comparison partner in the dominated context (rating) for Flixster-dominant, where a context (rating) is dominated by another one (social).

fer matrices for every context, to convert the general latent factors into a context-dependent latent factor. HETEROMF is particularly helpful for cold-start entities and contexts that are dominated by other contexts. The transfer matrices in HETEROMF are learned per context and prevent the learnt factors for a context from being dominated by other contexts. Also, for entities that are cold start in a context, HETEROMF uses the product of the base latent factor, learned from observations for *this* entity in *other* contexts, and the transfer matrix, learned from the observations for *all* entities in *this* context, to generate a proper context-specific latent factor.

We performed comprehensive experiments on two real datasets from Epinions and Flixster. Both data sets feature interactions between users and interactions between users and items (products or movies) in two different contexts. Experimental results demonstrate that HETEROMF substantially outperforms CMF, particularly for cold-start entities and for contexts where interactions in one context are dominated by those of other contexts. Additionally, our experimental analysis convincingly highlights the weaknesses of CMF exactly as we outlined in the technical sections, w.r.t. different kinds of users and different contexts.

This work suggests several interesting directions for future research. All interactions considered in this paper are binary, where a pair of entities (from the same or different types) are involved. There are real applications where the “arity” of interactions can

Table 2: Average running time of a single EM iteration in different datasets.

Dataset	HeteroMF	CMF
Epinions	34min.	7.5min.
Epinions-dominant	18min.	4min.
Flixster	220min.	48min.
Flixster-dominant	116min.	26min.

be higher. As an example, in Amazon, products are reviewed by reviewers, and both products and reviews may be rated by raters. Here, a rating on a review is really an interaction between three entity types – rater, reviewer and product. Tensor factorization models and their extensions have been used by Moghaddam et al. [11] for predicting helpfulness of reviews. It is interesting to enhance tensor factorization models to account for multiple interaction contexts. Another direction is incorporating entity attributes. For simplicity and generality, in this paper we do not assume access to entity attributes. However, the HETEROMF framework can take advantage of user demographics, movie genres and actors, and product specs, when available. Extending our experiments to incorporate these is interesting. Finally, we note that the features extracted in meta-path based approaches [20, 19] can also be used as features in HETEROMF.

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(6):734–749, 2005.
- [2] D. Agarwal, B.-C. Chen, and B. Pang. Personalized recommendation of user comments via factor models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP’11)*, pages 571–582, 2011.
- [3] R. M. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD’07: The 13th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, pages 95–104, 2007.
- [4] B.-C. Chen, J. Guo, B. Tseng, and J. Yang. User reputation in a comment rating environment. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’11)*, pages 159–167, 2011.
- [5] 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, RecSys’10*, pages 135–142, 2010.
- [6] Y. Koren. Factorization meets the neighborhood a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’08*, pages 426–434, 2008.
- [7] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’09*, pages 447–456, 2009.
- [8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42:30–37, 2009.

- [9] C. Lippert, S. H. Weber, Y. Huang, V. Tresp, M. Schubert, and H.-P. Kriegel. Relation prediction in Multi-Relational domains using matrix factorization. In *NIPS 2008 Workshop on Structured Input Structure Output*, 2008.
- [10] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'09, pages 203–210, 2009.
- [11] S. Moghaddam, M. Jamali, and M. Ester. Etf: extended tensor factorization model for personalizing prediction of review helpfulness. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 163–172, 2012.
- [12] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 61–70, 2002.
- [13] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Twenty-First Annual Conference on Neural Information Processing Systems*, NIPS'07, 2007.
- [14] C. Shi, X. Kong, P. S. Yu, S. Xie, and B. Wu. Relevance search in heterogeneous networks. In *Proceedings of the 15th International Conference on Extending Database Technology (EDBT'12)*, pages 180–191, 2012.
- [15] X. Shi, Q. Liu, W. Fan, Q. Yang, and P. S. Yu. Predictive modeling with heterogeneous sources. In *Proceedings of SIAM International Conference on Data Mining (SDM'10)*, 2010.
- [16] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. In *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM'10)*, pages 1049–1054, 2010.
- [17] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'08)*, pages 650–658, 2008.
- [18] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM '11)*, pages 121–128, 2011.
- [19] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *Proceedings of the fifth ACM international conference on Web search and data mining (WSDM'12)*, pages 663–672, 2012.
- [20] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *In 37th international conference on Very Large Databases (VLDB'11)*, 2011.
- [21] C. Wang, R. Raina, D. Fong, D. Zhou, J. Han, and G. Badros. Learning relevance from heterogeneous social network and its application in online targeting. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR'11)*, pages 655–664, 2011.
- [22] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 537–546, 2011.
- [23] X. Yu, Q. Gu, M. Zhou, and J. Han. Citation prediction in heterogeneous bibliographic networks. In *Proceeding of SIAM international conference on Data Mining (SDM'12)*, pages 1119–1130, 2012.
- [24] L. Zhang, D. Agarwal, and B.-C. Chen. Generalizing matrix factorization through flexible regression priors. In *Proceedings of the fifth ACM conference on Recommender systems (Recsys'11)*, pages 13–20, 2011.