Coupled Poisson Factorization Integrated with User/Item Metadata for Modeling Popular and Sparse Ratings in Scalable Recommendation

Trong Dinh Thac Do and **Longbing Cao**
Advanced Analytics Institute, University of Technology Sydney,
TrongDinhThac.Do@student.uts.edu.au and Longbing.Cao@uts.edu.au

Abstract

Modelling sparse and large data sets is highly in demand yet challenging in recommender systems. With the computation only on the non-zero ratings, Poisson Factorization (PF) enabled by variational inference has shown its high efficiency in scalable recommendation, e.g., modeling millions of ratings. However, as PF learns the ratings by individual users on items with the Gamma distribution, it cannot capture the coupling relations between users (items) and the rating popularity (i.e., favorable rating scores that are given to one item) and rating sparsity (i.e., those users (items) with many zero ratings) for one item (user). This work proposes *Coupled Poisson Factorization* (CPF) to learn the couplings between users (items), and the user/item attributes (i.e., metadata) are integrated into CPF to form the Metadata-integrated CPF (mCPF) to not only handle sparse but also popular ratings in very large-scale data. Our empirical results show that the proposed models significantly outperform PF and address the key limitations in PF for scalable recommendation.

Introduction

Recommender Systems (RS) play increasingly important roles in many applications, including online businesses and social media. Collaborative filtering is a basic method that has been widely explored in RS. For example, user-based collaborative filtering makes predictions about the interest of a user based on an analysis of the preferences of other similar users. As a fundamental tool for collaborative filtering, Matrix Factorization (MF) (Koren, Bell, and Volinsky 2009) has undergone many variations such as Non-negative Matrix Factorization (NMF) (Lee and Seung 1999).

However, as discussed in (Mnih and Salakhutdinov 2008; Gopalan, Hofman, and Blei 2015), MF models face significant challenges in handling real-life RS problems, e.g., they cannot handle large data because of the intensive mathematical computation required. Although different MF variants have been explored to address such issues, e.g., Probabilistic Matrix Factorization (Mnih and Salakhutdinov 2008) handles large amount of data based on its statistical method, they are still inefficient especially for sparse data, since they perform the computation on all data which usually consists of many zero ratings. Accordingly, Gopalan, Hofman, and Blei (2015) introduced Poisson Factorization (PF) and used the variational inference method for scalable recommendation on large and sparse data by only scanning non-zero ratings, which appears to be very promising.

PF factorizes the ratings by individual users on items based on their posterior assumption of Gamma distribution, but it may not capture the rating popularity and sparsity and comprehensive user/item couplings embodied through explicit and implicit variables (Cao, Ou, and Yu 2012; Cao 2015), which essentially shows recommendation problems are non-IID (Cao 2016). For example, the complete conditional posterior of items is in Eq. (1). a is the initial *shape* of the Gamma distribution-based item weights β_{ik}; $a + \sum_u z_{uik}$ is the *shape* of the Gamma distribution-based item weights after we sample the observed data; and z_{uik} represents the rating given by user u on item i.

$$\beta_{ik}|\theta, \eta, z, y \sim Gamma(a + \sum_u z_{uik}, \eta_i + \sum_u \theta_{uk}) \quad (1)$$

Let us illustrate the PF problems using the toy examples in Table 1a by estimating the weight of the movie 'The Game (1997)' for user 940. With the posterior distribution in Eq. (1), the result is around 2 or 3. However, as shown in Table 1a, many users gave ratings 1 to 'The Game (1997)', hence it may be more practical to set the item weight closer to 1. This is the problem of *rating popularity* for a user (item).

PF is good at capturing sparse ratings using the Gamma distribution (Gopalan, Hofman, and Blei 2015) as in Eq. (1). Only the non-zero ratings are added to fit the *shape* of the Gamma distribution. With sparse ratings for a user (item), most of the weights will be 0 and only a few will be larger than 0. This makes the computation faster but may lead to wrong recommendations. We explain this scenario using Table 1b. It may be more reasonable to recommend 'The Game (1997)' to user 405 since two users rated it as 5. However, PF only suggests 2 or 3. This problem appears when many users give high ratings to an item while others do not rate it. This creates the problem of *rating sparsity* for a user (item).

In this work, we address the aforementioned rating popularity and sparsity issues in PF-based RS by involving and modeling user/item metadata and the user and item couplings (Cao 2015; 2016). First, the Coupled Poisson Factorization (CPF) is proposed to model user/item relations. Here "Coupled" refers to (1) *coupled users* by learning the

Table 1: Rating Popularity and Sparsity Examples

	The Game (1997)	Scream (1996)	Air Force One (1997)	Groundhog Day (1993)
179	5	5	4	4
193	1	?	4	?
204	1	3	3	?
15	1	?	?	?
458	1	3	?	5
626	1	3	?	4
940	?	?	5	4

(a) Popularity within an item

	The Game (1997)	Scream (1996)	Air Force One (1997)	Groundhog Day (1993)
179	5	5	4	4
91	5	?	4	?
392	?	4	2	?
263	?	?	3	4
286	?	?	?	4
324	?	5	?	?
405	?	5	?	4

(b) Sparsity within an item

relations between the ratings of two users on one item (e.g., in Table 1a, the rating by user 193 is 1 which is similar to the rating given by user 204 but different (5) to that given by user 179); (2) *coupled items* by learning the relations between the ratings on two items given by the same user (e.g., user 179 gave a similar rating (5) for 'The Game (1997)' and 'Scream (1996)' but a different rating (1) for 'Starship Troopers (1997)'). CPF factorizes the relations between users (items) to obtain their weights. It only calculates the weights based on similar ratings and therefore inherits the strength of PF on a sparse matrix. Furthermore, CPF models the rating behavior similarity of users on items and thus can address rating sparsity and popularity. As a result, CPF sets the weight of 'The Game (1997)' by user 940 to 1 in Table 1a and that by 405 to 5 in Table 1b. Second, we integrate user/item metadata into CPF to generate a new model called the *Metadata-integrated Coupled Poisson Factorization* (mCPF). Since the sparsity of rating data is often very high in real data, it is natural to integrate user/item metadata to RS (Cao 2016). For example, the 'genre' of a movie and the 'occupation' of a user are integrated with their ratings.

CPF and mCPF own the following properties. First, built on the PF strength, CPF and mCPF need only to scan the non-zero ratings. Hence, CPF and mCPF can handle massive sparse data in recommendation. Second, CPF captures the user/item interactions by factorizing the matrix of user/item similarity w.r.t. the ratings, as shown by the examples in Tables 1a and 1b. As a result, CPF addresses rating popularity and sparsity for one user/item. In addition, by incorporating user/item metadata, mCPF further improves the recommendation precision, especially when the sparsity of data is extremely high. Lastly, as CPF and mCPF are fully Bayesian and conjugate models, variational inference can be applied for scalable inference on a large and sparse matrix.

The CPF and mCPF Models

Coupled User Poisson Factorization (CuPF)

Real-life data includes the matrix of ratings given by users to items, Y, where y_{ui} is the rating by user u to item i (e.g., from 1 to 5) and 0 if there is no rating. Typically, this kind of observed data is highly sparse.

In CuPF, we first transform the ratings to the matrix of similar ratings between users (*coupled users*), SU. The elements of similarity matrix SU are defined as: $SU_{uv,i} = 1$ if the rating by user u to item i (i.e., y_{ui}) is similar to the rating by user v to item i (y_{vi}), and 0 otherwise. In practice, for examples like the ratings which scored from 1 to 5 in Movielens, there may be varied ways to set the similarity. Our experiments show the best results by setting $SU_{uv,i} = 1$ when $|y_{ui} - y_{vi}| \leq 1$.

Built on the strength of Poisson Factorization (Gopalan, Hofman, and Blei 2015), the similarity matrix is then factorized by using Poisson distribution as in (Canny 2004) to the vector of K latent feature for each item, β_{ik} and the vector of K latent rating preference similarity for each coupled user θ_{uvk} (instead of the vector of K latent preferences for each user as in PF). The Gamma distribution is given to β_{ik}, θ_{uvk} similar to (Gopalan, Hofman, and Blei 2015) and the Poisson distribution ($SU_{uv,i} \sim Poisson(\sum_k \theta^T_{uvk}\beta_{ik})$ for the observed similar ratings of users. CuPF still keeps the feature of heterogeneity across users and items as discussed in (Koren, Bell, and Volinsky 2009) by placing the additional Gamma prior on the item's latent attractiveness η_i and latent behavior similarity of coupled users ξ_{uv} (instead of the user's activity in PF). Hence, it can capture the sparse representation of items and the relations of users.

The CuPF's generative process is as follows.

(1) For each relation between users u and v:

(a) Sample latent behavior similarity:

$$\xi_{uv} \sim Gamma(a', b') \qquad (2)$$

(b) Sample latent preference similarity for each component k:

$$\theta_{uvk} \sim Gamma(a, \xi_{uv}) \qquad (3)$$

(2) For each item i:

(a) Sample latent attractiveness:

$$\eta_i \sim Gamma(c', d') \qquad (4)$$

(b) Sample latent feature for each component k:

$$\beta_{ik} \sim Gamma(c, \eta_i) \qquad (5)$$

(3) For each relation between users u and v and each item i, sample rating similarity:

$$SU_{uv,i} \sim Poisson(\sum_k \theta^T_{uvk}\beta_{ik}) \qquad (6)$$

The missing rating similarity (i.e., zeros) between users u and v to item i are estimated by $SU_{uv,i} = 1$ if the expected Poisson parameter $E[\theta^T_{uvk}\beta_{ik}] \geq \epsilon$; and 0 otherwise. The missing ratings by user u to item i are then recovered by its rating similarity with other users to item i by taking the rating popularity into account.

$$y_{ui} = \frac{\sum_v y_{vi}\delta_{vi}SU_{u,i}}{\sum_v SU_{uv,i}} \tag{7}$$

where δ_{vi} is the popularity of the rating given by user v to item i (y_{vi}) and is defined as the fraction of the number of users with rating y_{vi} of all users that rated item i.

For example, the popularity of rating 1 by users 193, 204, 15, 458 and 626, which is $5/6$ as shown in Table 1a, may be higher than that of rating 5 by user 179, which is $1/6$. By doing this, the post-processing using Eq. (7) can capture more information as described in Tables 1a and 1b, which estimates the ratings close to 1 for user 940 and 5 for user 405. We show the graphical model of CuPF in Figure 1a.

Coupled Item Poisson Factorization (CiPF)

Similar to CuPF, CiPF factorizes the observed matrix of similar ratings between items SI. The elements of similarity matrix SI are defined as: $SI_{u,ij} = 1$ if the rating by user u to item i (i.e., y_{ui}) is similar to the rating by user u to item j (y_{uj}); and 0 otherwise. We place the Poisson distribution for SI and the Gamma prior for the vector of K latent preferences θ_{uk} for each user and the vector of K latent feature similarity β_{ijk} for each coupled item (instead of the vector of K latent attributes for each item in PF). The Gamma distribution is also given to the user's latent behavior ξ_u and the item's latent attractiveness similarity η_{ij}.

The CiPF's generative process is as follows.

(1) For each user u:

(a) Sample latent behavior:

$$\xi_u \sim Gamma(a', b') \tag{8}$$

(b) Sample latent preference of each component k:

$$\theta_{uk} \sim Gamma(a, \xi_u) \tag{9}$$

(2) For each relation between items i and j:

(a) Sample latent attractiveness similarity:

$$\eta_{ij} \sim Gamma(c', d') \tag{10}$$

(b) Sample latent feature similarity for each component k:

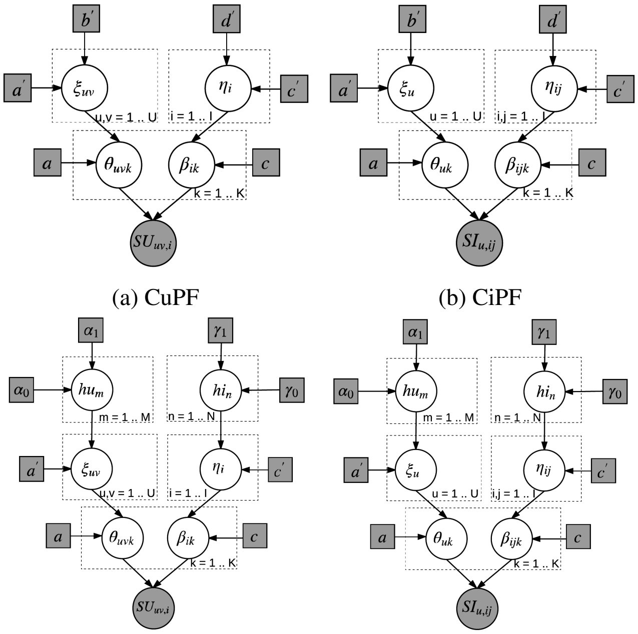
$$\beta_{ijk} \sim Gamma(c, \eta_{ij}) \tag{11}$$

(3) For each user u and relation between items i and j, sample rating similarity:

$$SI_{u,ij} \sim Poisson(\sum_k \theta_{uk}^T \beta_{ijk}) \tag{12}$$

The missing rating similarity (i.e., zeros) given by user u to items i and j are estimated by $SI_{u,ij} = 1$ if the expected Poisson parameter $E[\theta_{uk}^T \beta_{ijk}] \geq \epsilon$; and 0 otherwise. The missing ratings by user u to item i are then recovered according to its rating similarity to other items j by taking the rating popularity into account.

$$y_{ui} = \frac{\sum_j y_{uj}\delta_{uj}SI_{u,ij}}{\sum_j SI_{u,ij}} \tag{13}$$



(a) CuPF

(b) CiPF

(c) mCuPF

(d) mCiPF

Figure 1: Metadata-integrated Coupled Poisson Factorization.

where δ_{uj} is the popularity of ratings by user u to item j (i.e., y_{uj}) and is defined as the fraction of the number of items with ratings y_{uj} in all items that have the same ratings as user u. Figure 1b shows the graphical model of CiPF.

Next, we introduce the process of integrating the metadata of users and items to CuPF and CiPF.

Integrating Metadata to Coupled User Poisson Factorization (mCuPF)

For each user attribute m in the metadata, sample the weight:

$$hu_m \sim Gamma(\alpha_0, \alpha_1) \tag{14}$$

For each item attribute n in the metadata, sample the weight:

$$hi_n \sim Gamma(\gamma_0, \gamma_1) \tag{15}$$

For each latent behavior similarity of coupled users u and v:

$$\xi_{uv} \sim Gamma(a', \prod_{m=1}^{M} hu_m^{fu_{u,m}fu_{v,m}}) \tag{16}$$

For each item i's latent attractiveness:

$$\eta_i \sim Gamma(c', \prod_{n=1}^{N} hi_n^{fi_{i,n}}) \tag{17}$$

We apply the Gamma prior to the weight of each user attribute, hu_m, e.g., the 'age' of a user, as in Eq. (14). The weight of user attribute hu_m only affects the behavior similarity of users ξ_{uv} and further affects the preference similarity of users θ_{uvk} if and only if $fu_{u,m} = fu_{v,m} = 1$ as in Eq. (16). That means both users u and v have the attribute m.

hu_m measures the degree of influence of each user attribute. For example, the 'occupation' of a user may have less influence than the 'age' of a user in Movielens. The weight of an item attribute hi_n, e.g., the 'genre' of a movie, is also given a Gamma distribution as in Eq. (15). The weight of item attribute hi_n only affects the item's latent attractiveness η_i when item i has the attribute n (i.e., $fi_{i,n} = 1$). The graphical model of mCiPF is shown in Figure 1c. In the section about the model properties, we explain why they are integrated in this way.

Integrating Metadata to Coupled Item Poisson Factorization (mCiPF)

For each user u's latent behavior:

$$\xi_u \sim Gamma(a', \prod_{m=1}^{M} hu_m^{fu_{u,m}}) \tag{18}$$

For the attractiveness similarity between items i and j:

$$\eta_{ij} \sim Gamma(c', \prod_{n=1}^{N} hi_n^{fi_{i,n} fi_{j,n}}) \tag{19}$$

Similar actions are taken on mCiPF as on mCuPF with the user attribute hu_m in Eq. (14) and item attribute hi_n in Eq. (15). This time, we integrate hu_m to individual user and hi_n to the relations between items as in Eqs. (18) and (19). The graphical model of mCiPF is shown in Figure 1d.

Inference for CPF/mCPF

Using CPF/mCPF for recommender systems depends on solving the posterior inference problem. We apply the mean-field variational inference (VI) to our models as it is efficient for large-scale probabilistic models (Wainwright, Jordan, and others 2008), compared to other sampling approaches like MCMC (Gilks, Richardson, and Spiegelhalter 1995). Using VI, we find the family of distributions over the hidden variables and the members of this family by tuning the parameters to minimize the Kullback-Leibler (KL) divergence to the true posterior. Due to space limitations, we only introduce the VI method for CuPF/mCuPF, which is similiar to that for CiPF/mCiPF.

Variational Inference for CuPF

Given SU, we compute the posterior distributions of the latent preference similarity of coupled users θ_{uvk}, the item's latent feature β_{ik}, the latent behavior similarity of coupled users ξ_{uv}, and the item's latent attractiveness η_i. The same approach as in (Gopalan, Hofman, and Blei 2015) is taken here, however we replace the similarity between ratings by coupled users u, v to item i, $SU_{uv,i}$, with auxiliary latent variable $w_{uv,i} \sim Poisson(\theta_{uvk}\beta_{ik})$. Accordingly, the similarity SU is expressed as follows:

$$SU_{uv,i} = \sum_k w_{uv,i,k} \tag{20}$$

Similar to (Gopalan, Hofman, and Blei 2015), the inference only considers $w_{uv,i}$, $SU_{uv,i} = 1$. The mean-field family assumes each distribution is independent of the others.

$$\begin{aligned} q(\theta, \beta, \xi, \eta, w) = \prod_{uv,k} q(\theta_{uvk}|\nu_{uvk}) \prod_{i,k} q(\beta_{ik}|\mu_{ik}) \\ \prod_{uv} q(\xi_{uv}|\kappa_{uv}) \prod_{i} q(\eta_i|\tau_i) \prod_{uv,i,k} q(w_{uv,i,k}|\phi_{uv,i,k}) \end{aligned} \tag{21}$$

We use the class of conditionally conjugate priors for θ_{uvk}, β_{ik}, ξ_{uv}, η_i and $w_{uv,i}$ to update the variational parameters $\{\nu, \mu, \kappa, \tau, \phi\}$. For the Gamma distribution, we update both hyper-parameters: *shape* and *rate*.

(1) Update *shape* and *rate* of κ_{uv}:

$$\kappa_{uv,0} = a' + Ka \tag{22}$$

$$\kappa_{uv,1} = b' + \sum_k \frac{\nu_{uvk,0}}{\nu_{uvk,1}} \tag{23}$$

(2) Update *shape* and *rate* of τ_i:

$$\tau_{i,0} = c' + Kc \tag{24}$$

$$\tau_{i,1} = d' + \sum_k \frac{\mu_{ik,0}}{\mu_{ik,1}} \tag{25}$$

(3) Update $\phi_{uv,i,k}$:

$$\begin{aligned} \phi_{uv,i,k} =& exp\{\Psi(\nu_{uvk,0}) - log(\nu_{uvk,1}) \\ &+ \Psi(\mu_{ik,0}) - log(\mu_{ik,1})\} \end{aligned} \tag{26}$$

where $\Psi()$ is the *digamma* function.

(4) Update *shape* and *rate* of ν_{uvk}:

$$\nu_{uvk,0} = a + \sum_i SU_{uv,i}\phi_{uv,i,k} \tag{27}$$

$$\nu_{uvk,1} = \frac{\kappa_{uv,0}}{\kappa_{uv,1}} + \sum_i \frac{\mu_{ik,0}}{\mu_{ik,1}} \tag{28}$$

(5) Update *shape* and *rate* of μ_{ik}:

$$\mu_{ik,0} = c + \sum_{u,v} SU_{uv,i}\phi_{uv,i,k} \tag{29}$$

$$\mu_{ik,1} = \frac{\tau_{i,0}}{\tau_{i,1}} + \sum_{u,v} \frac{\nu_{uvk,0}}{\nu_{uvk,1}} \tag{30}$$

Owing to the limited space, the details of the deviation, which is similar to PF, are ignored here. We will instead give details of the process of deviation of integrating user/item metadata to CuPF in the following section.

Integrating User/Item Metadata to Coupled User Poisson Factorization (mCuPF)

In mCuPF, we further learn two more parameters ζ_m and ρ_n for the weight of user attributes and item attributes by minimizing the KL divergence to the posterior.

$$\begin{aligned} q(\zeta, \rho, \theta, \beta, \xi, \eta, w) = \prod_m q(hu_m|\zeta_m) \prod_n q(hi_n|\rho_n) \\ \prod_{uv,k} q(\theta_{uvk}|\nu_{uvk}) \prod_{i,k} q(\beta_{ik}|\mu_{ik}) \prod_{uv} q(\xi_{uv}|\kappa_{uv}) \\ \prod_i q(\eta_i|\tau_i) \prod_{uv,i,k} q(w_{uv,i,k}|\phi_{uv,i,k}) \end{aligned} \tag{31}$$

With the Gamma distribution in Eqs. (14) and (16),

$$p(hu_m|\alpha_0, \alpha_1) \propto hu_m^{\alpha_0 - 1} exp\{-\alpha_1 hu_m\} \tag{32}$$

$$p(\xi_{uv}|a', hu_m) \propto \\ (\prod_{m=1}^{M} hu_m^{f_{u_{u,m}} f_{u_{v,m}} a'}) exp\{-(\prod_{m=1}^{M} hu_m^{f_{u_{u,m}} f_{u_{v,m}}}) \xi_{uv}\} \tag{33}$$

The posterior probability of weight hu_m becomes:

$$p(hu_m|\alpha_0, \alpha_1, \xi_{uv}) \propto p(hu_m|\alpha_0, \alpha_1) \prod_{u,v} p(\xi_{uv}|a', hu_m) \\ \propto hu_m^{\alpha_0 + \aleph_{uv,m} a' - 1} exp\{-(\alpha_1 + \sum_{u,v} \xi_{uv}) hu_m\} \tag{34}$$

$\aleph_{uv,m}$ is the number of coupled users u, v both having attribute m. The posterior Gamma distribution of hu_m is

$$hu_m \sim Gamma(\alpha_0 + \aleph_{uv,m} a', \alpha_1 + \sum_{u,v} \xi_{uv}) \tag{35}$$

hu_m is affected by $\aleph_{uv,m}$ and the weight of the relations between users based on the ratings (i.e., ξ_{uv}). Similarly, the posterior distribution for the weight of item attribute, hi_n, is

$$hi_n \sim Gamma(\gamma_0 + \chi_{i,n} c', \gamma_1 + \sum_{i} \eta_i) \tag{36}$$

where $\chi_{i,n}$ is the number of items i that have attribute n.

After obtaining the posterior distribution for the Gamma distribution of hu_m and hi_n, we update *shape* and *rate* of the variational parameters as follows.

(1) For coupled users u and v:

$$\zeta_{m,0} = \alpha_0 + \aleph_{uv,m} a' \tag{37}$$

$$\zeta_{m,1} = \alpha_1 + \sum_{u,v} \kappa_{uv,0} / \kappa_{uv,1} \tag{38}$$

(2) For item i:

$$\rho_{n,0} = \gamma_0 + \chi_{i,n} c' \tag{39}$$

$$\rho_{n,1} = \gamma_1 + \sum_{i} \tau_{i,0} / \tau_{i,1} \tag{40}$$

Compared to CuPF, since the *rate* of the Gamma distribution of latent behavior similarity of coupled users u and v, ξ_{uv}, and the latent attractiveness of the item η_i have been changed as in Eqs. (16) and (17), we change the update of Eqs. (23) and (25) as follows.

$$\kappa_{uv,1} = \prod_{m=1}^{M} (\zeta_{m,0}/\zeta_{m,1})^{f_{u_{u,m}} f_{u_{v,m}}} + \sum_{k} \frac{\nu_{uvk,0}}{\nu_{uvk,1}} \tag{41}$$

$$\tau_{i,1} = \prod_{n=1}^{N} (\rho_{n,0}/\rho_{n,1})^{f_{i_{i,n}}} + \sum_{k} \frac{\mu_{ik,0}}{\mu_{ik,1}} \tag{42}$$

By the mean-field variational inference, the coordinate ascent is used to iteratively optimize each variational parameter while holding the others fixed (Jordan et al. 1999). The variational inference of mCuPF is listed in Algorithm 1 and its full process is shown in Algorithm 2.

Algorithm 1 Variational Inference for mCuPF

```
Initialize the variational parameters $\{\zeta, \rho, \nu, \mu, \kappa, \tau, \phi\}$.
Sample shape of latent behavior similarity of coupled users, $\xi_{uv}$, and shape of item's latent attractiveness, $\eta_i$, as in Eqs. (22) and (24).
Sample shape of the weight of user's attribute (in metadata), $hu_m$, and shape of the weight of item's attribute (in metadata), $hi_n$, as in Eqs. (37) and (39).
repeat
    for each rating similarity between coupled users $u, v$ to item $i$ that $SU_{uv,i} = 1$ do
        Update the multinominal as in Eq. (26).
    end for
    for each coupled users do
        Update the latent preference similarity as in Eqs. (27) and (28)
        Update rate of latent behavior similarity as in Eq. (41).
        for each user attribute in metadata do
            Update rate of the weight as in Eq. (38)
        end for
    end for
    for each item do
        Update the latent feature as in Eqs. (29) and (30).
        Update rate of latent attractiveness as in Eq. (42).
        for each item attribute do
            Update rate of the weight as in Eq. (40).
        end for
    end for
until convergence
```

Algorithm 2 mCuPF

```
Input: Rating matrix $Y$.
Output: Estimated missing ratings (i.e., zeros in $Y$).
Pre-Processing to get the similarity matrix $SU$.
Inference to optimize all parameters as in Algorithm 1.
Post-Processing as in (7) to get missing ratings.
```

Properties of CPF/mCPF and Related Work

We explain the properties of CPF/mCPF in the context of the related work. As discussed in the introduction, the proposed models CPF/mCPF hold the following four properties.

(1) CPF/mCPF are fast for a sparse matrix. The rating matrix is often sparse. When we transform it to a similarity matrix (i.e., coupled users SU and coupled items SI), the similarity matrix is also sparse, as we only keep the similarity pair ($SU_{uv,i} = 1$) but ignore the dissimilarity pair and zero elements ($SU_{uv,i} = 0$). For example, in CuPF, given the vector of latent preferences of coupled user u and v, θ_{uv}, and the vector of the item's latent features β_i, the probability based on Poisson distribution of the similarity of ratings by coupled users u and v to item i, $SU_{uv,i}$, is given below.

$$p(SU_{uv,i}|\theta_{uv}, \beta_i) = \frac{(\theta_{uv}^T \beta_i)^{SU} exp\{-\theta_{uv}^T \beta_i\}}{SU_{uv,i}!} \tag{43}$$

The corresponding log probability of similar matrix SU

is given as follows.

$$
\begin{aligned}
& log(SU|\theta,\beta) = \\
& (\sum_{SU_{uv,i}\neq 0} SU_{uv,i} log(\theta_{uv}^T \beta_i) - log(SU_{uv,i}!)) \\
& -(\sum_{u,v} \theta_{uv})^T (\sum_i \beta_i)
\end{aligned}
\tag{44}
$$

When $SU_{uv,i} = 0$ (i.e., $log(SU_{uv,i}!) = 0$), it will not affect the log probability. This feature is inherited from the Poisson Factorization. It does not require optimization techniques to reduce the computational time as in the classical Matrix Factorization (Dror et al. 2012; Mairal et al. 2010).

(2) CPF and mCPF better capture the characteristics in real data, especially for the examples in Tables 1a and 1b. This is because we separate the process of calculating the weight of the rating by user u to item i by taking the rating popularity and sparsity for one user/item into account as in Eqs. (7) and (13). If we do not care about the popularity (i.e., set all as 1), we have the results as in PF.

(3) mCPF achives improved precision by integrating user/item metadata. Given the Gamma distribution of relations between users in Eq. (16), we can separate it into two elements: (1) *shape* a' and (2) *rate* $\prod_{m=1}^{M} hu_m^{fu_{u,m} fu_{v,m}}$. When the *shape* is fixed, the larger the *rate*, the higher the weight (probability) we sample from the distribution. Hence, if two users have the same attribute m with weight hu_m (e.g., the same 'age'), it causes the *rate* of the Gamma distribution in Eq. (16) to increase. Consequently, it will increase the probability of the weight of the relation between users. Further, we do not have to fix hu_m since it can be easily learned from the Gamma-Gamma conjugate.

In the literature, several studies tried to integrate the document-word matrix into the Poisson Factorization (Acharya et al. 2015; Gopalan, Charlin, and Blei 2014; Zhang and Wang 2015; Hu, Rai, and Carin 2016). Different from these models, mCPF incorporates the metadata with general attributes (e.g., the categorical attributes; not just the text data). Recent work in (Zhao, Du, and Buntine 2017; Fan et al. 2017) tried to integrate the general attributes into probabilistic models for link prediction, which works on small data due to the limitation of the applied Gibbs sampling.

(4) The variational inference for CPF and mCPF applies to massive data. Variational inference has proved to be efficient for probabilistic models that involve a large amount of data. As mCPF is built on the Gamma-Gamma-Gamma-Poisson distribution, it is fully Bayesian and conjugate. As discussed in (Ghahramani and Beal 2001; Hoffman et al. 2013), we can easily build a variational algorithm for fully Bayesian and conjugate models.

Empirical Results

Baseline methods To the best of our knowledge, no existing methods have incorporated user/item metadata into the Poisson Factorizaton. As shown in (Gopalan, Hofman, and Blei 2015), the hierarchical PF (HPF) outperforms baseline models including basic PF, NMF, Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003), and PMF. Due to space limitations, we only show the comparison with HPF.

Datasets Few datasets are available with metadata. CPF/mCPF are tested on four public datasets available with massive ratings and some metadata.

(1) Movielens100K (Harper and Konstan 2016) includes user demographic information 'age', 'gender', 'occupation' and 'zip'. Here 'age' is chosen from the ranges: $1 \rightarrow$ "$Under18$", $18 \rightarrow$ "$18-24$", $25 \rightarrow$ "$25-34$", $35 \rightarrow$ "$35-44$", $45 \rightarrow$ "$45-49$", $50 \rightarrow$ "$50-55$", $56 \rightarrow$ "$56+$". The item metadata includes the 'genre' (e.g., 'action', 'adventure', 'animation', ...), 'release date', and 'video release date' of movies.

(2) Movielens1M includes the same metadata as in Movielens100k.

(3) Movielens10M contains the 'genre' of the movies.

(4) Book-Crossing (Ziegler et al. 2005) contains user demographic information 'location' and 'age', and we also encode 'age' in the same way as in Movielens100K. The book information includes 'book title', 'book author', 'year of publication', and 'publisher'.

Parameter settings The same settings for HPF are applied to CPF/mCPF for fair comparison. We set $a = c = a' = b' = c' = d' = 0.3$. Further, the hyperparameters of user and item attributes, α_0, α_1 and γ_0, γ_1, are set to 0.1. The number of latent variables K is set to 100.

Evaluation method We use 20% of the ratings data for testing and 80% for training. Movielens100K and Movielens1M have been divided into testing and training. In Book-Crossing, we randomly extract data to form the testing and training sets. We get the top-N recommendations in the training set with the highest prediction score as in Eq. (7) for CuPF/mCuPF and in Eq. (13) for CiPF/mCiPF. In the testing, we compute the precision-at-N, which measures the fraction of relevant items in a users top-N recommendations. We compute recall-at-N, which is the fraction of the testing items that are present in the top-N recommendations.

Convergence We measure the convergence by computing the prediction accuracy on the validation set that is extracted by randomly selecting 1% of the ratings in the training set.

Predicting top-N recommendations Figure 2 shows the normalized mean precision and normalized mean recall of the top-20 recommendations made by CPF/mCPF, which are consistent with the results of $N \neq 20$, omitted due to space limitations. CPF and mCPF outperform HPF in both datasets with up to 11% improvement of mean precision on Book-Crossing. The significant improvement made by mCuPF and mCiPF results from metadata integration. The results on Movielens10M are less significant due to the insufficient (only the 'genre' of the movies) metadata.

Addressing rating popularity and sparsity Figure 3 reports the popularity level and sparsity level for the top-100 recommendations on Movielens1M and Book-Crossing. The *popularity level* is the number of recommended items (users) with greater than 50% low ratings (1 and 2) out of the total ones for one user (item). The *sparsity level* is the number of recommended items (users) with greater than 50% high (4 and 5) ratings of the total non-zero ones for

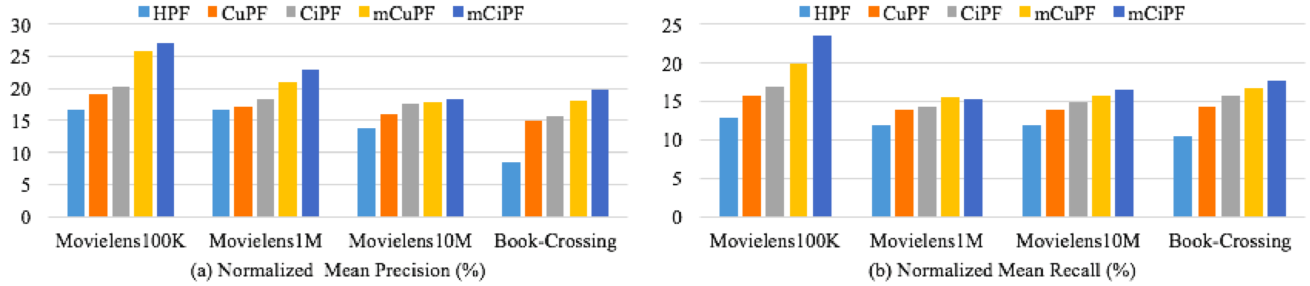
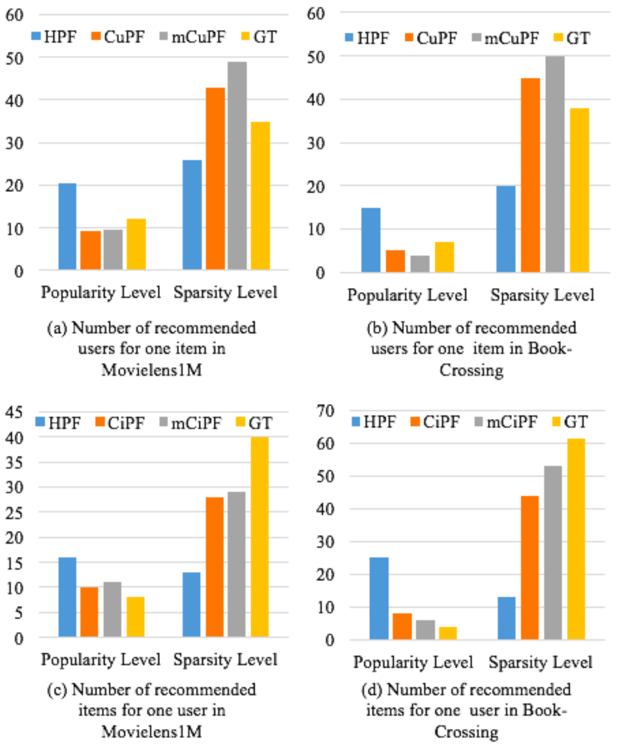


Figure 2: Predictive Performance of Top 20 Recommendations on Four Datasets.



(a) Number of recommended users for one item in Movielens1M

(b) Number of recommended users for one item in Book-Crossing

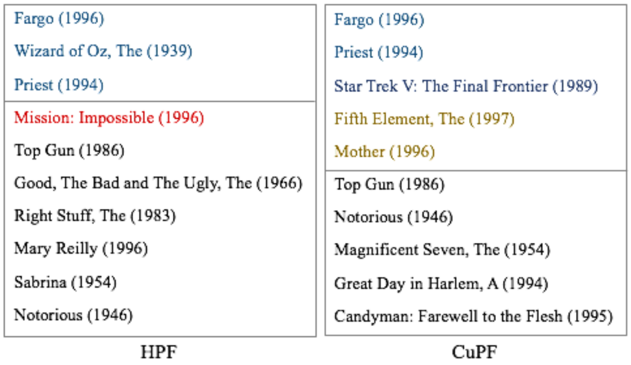
(c) Number of recommended items for one user in Movielens1M

(d) Number of recommended items for one user in Book-Crossing

Figure 3: Rating Popularity and Sparsity Test.

one user (item). Such high ratings are sparse compared to greater than 50% of the total users (items) having missing ratings. We report the normalized mean for all users/items. It shows HPF has high popularity but low sparsity levels, indicating high false recommendations of popular low ratings but low true recommendations of sparse high ratings. In contrast, CPF and mCPF have low popularity but high sparsity levels, which are much more consistent with the GT values corresponding to the actual (ground-truth) mean popularity and sparsity levels in the test data. This explains why our models work better than HPF.

Case studies Figure 4 illustrates the top-10 recommendations for user 184 in Movielens100K. HPF hits three (in blue) correctly while recommending seven (in black) wrong ones including item ‘Mission: Impossible (1996)’ (in red) which has a large number of low ratings (585 out of a total of 943 users who rated it 1 and 2). CuPF hits five correctly including ‘Fifth Element, The (1997)’ and ‘Mother (1996)’ (in yellow), each has a good number of high ratings but is relatively sparse, since ‘Fifth Element, The (1997)’ received ratings 4 and 5 by 88 out of a total of 943 users (incl. 122 non-zero ratings) and ‘Mother (1996)’ received ratings 4 and 5 by 76 users (with 125 non-zero ones). This example shows CPF makes more accurate recommendations especially on items with sparse high ratings, while HPF wrongly recommends items with popular low ratings.



HPF	CuPF
Fargo (1996)	Fargo (1996)
Wizard of Oz, The (1939)	Priest (1994)
Priest (1994)	Star Trek V: The Final Frontier (1989)
Mission: Impossible (1996)	Fifth Element, The (1997)
Top Gun (1986)	Mother (1996)
Good, The Bad and The Ugly, The (1966)	Top Gun (1986)
Right Stuff, The (1983)	Notorious (1946)
Mary Reilly (1996)	Magnificent Seven, The (1954)
Sabrina (1954)	Great Day in Harlem, A (1994)
Notorious (1946)	Candyman: Farewell to the Flesh (1995)

Figure 4: Top-10 Recommendations for User 184.

Conclusions

While Poisson Factorization with variational inference significantly outperforms Matrix Factorization and LDA etc. in addressing sparse data for scale recommendation, it fails in modeling popular and sparse ratings. Accordingly, we propose the Coupled Poisson Factorization (CPF) to learn the relations between users and items and the Metadata-integrated CPF (mCPF) to integrate user/item metadata. CPF and mCPF are the first models in the PF family that inherit the advantages of PF in handling sparse data and model the metadata and the rating relations between users (items) to address rating popularity and sparsity issues. Both CPF and mCPF significantly outperform its baseline PF and MF and LDA, for which the comparison results are omitted due to space limitations; our ongoing efforts are on developing more effective methods for tuning the parameters.

References

Acharya, A.; Teffer, D.; Henderson, J.; Tyler, M.; Zhou, M.; and Ghosh, J. 2015. Gamma process Poisson factorization for joint modeling of network and documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 283–299. Springer.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3(Jan):993–1022.

Canny, J. 2004. Gap: a factor model for discrete data. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 122–129. ACM.

Cao, L.; Ou, Y.; and Yu, P. S. 2012. Coupled behavior analysis with applications. *IEEE TKDE* 24(8):1378–1392.

Cao, L. 2015. Coupling learning of complex interactions. *J. Information Processing and Management* 51(2):167–186.

Cao, L. 2016. Non-iid recommender systems: A review and framework of recommendation paradigm shifting. *Engineering* 2(2):212 –224.

Dror, G.; Koenigstein, N.; Koren, Y.; and Weimer, M. 2012. The yahoo! music dataset and kdd-cup11. In *Proceedings of KDD Cup 2011*, 3–18.

Fan, X.; Da Xu, R. Y.; Cao, L.; and Song, Y. 2017. Learning nonparametric relational models by conjugately incorporating node information in a network. *IEEE Transactions on Cybernetics* 47(3):589–599.

Ghahramani, Z., and Beal, M. J. 2001. Propagation algorithms for variational bayesian learning. In *Advances in Neural Information Processing Systems*, 507–513.

Gilks, W. R.; Richardson, S.; and Spiegelhalter, D. 1995. *Markov chain Monte Carlo in practice*. CRC press.

Gopalan, P. K.; Charlin, L.; and Blei, D. 2014. Content-based recommendations with Poisson Factorization. In *Advances in Neural Information Processing Systems*, 3176–3184.

Gopalan, P.; Hofman, J. M.; and Blei, D. M. 2015. Scalable Recommendation with Hierarchical Poisson Factorization. In *Conference on Uncertainty in Artificial Intelligence*, 326–335.

Harper, F. M., and Konstan, J. A. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.

Hoffman, M. D.; Blei, D. M.; Wang, C.; and Paisley, J. 2013. Stochastic variational inference. *The Journal of Machine Learning Research* 14(1):1303–1347.

Hu, C.; Rai, P.; and Carin, L. 2016. Topic-based embeddings for learning from large knowledge graphs. In *Artificial Intelligence and Statistics*, 1133–1141.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine learning* 37(2):183–233.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8).

Lee, D. D., and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788.

Mairal, J.; Bach, F.; Ponce, J.; and Sapiro, G. 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11(Jan):19–60.

Mnih, A., and Salakhutdinov, R. R. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 1257–1264.

Wainwright, M. J.; Jordan, M. I.; et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1(1–2):1–305.

Zhang, W., and Wang, J. 2015. A collective Bayesian Poisson Factorization model for cold-start local event recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 1455–1464. ACM.

Zhao, H.; Du, L.; and Buntine, W. 2017. Leveraging node attributes for incomplete relational data. *arXiv preprint arXiv:1706.04289*.

Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, 22–32. ACM.