Metadata-dependent Infinite Poisson Factorization for Efficiently Modelling Sparse and Large Matrices in Recommendation

Trong Dinh Thac Do[†], **Longbing Cao**[†]
[†] Advanced Analytics Institute, University of Technology Sydney, Australia
trongdinhthac.do@student.uts.edu.au, longbing.cao@uts.edu.au

Abstract

Matrix Factorization (MF) is widely used in Recommender Systems (RSs) for estimating missing ratings in the rating matrix. MF faces major challenges of handling very sparse and large data. Poisson Factorization (PF) as an MF variant addresses these challenges with high efficiency by only computing on those non-missing elements. However, ignoring the missing elements in computation makes PF weak or incapable for dealing with columns or rows with very few observations (corresponding to sparse items or users). In this work, Metadata-dependent Poisson Factorization (MPF) is invented to address the user/item sparsity by integrating user/item metadata into PF. MPF adds the metadata-based observed entries to the factorized PF matrices. In addition, similar to MF, choosing the suitable number of latent components for PF is very expensive on very large datasets. Accordingly, we further extend MPF to Metadata-dependent Infinite Poisson Factorization (MIPF) that integrates Bayesian Nonparametric (BNP) technique to automatically tune the number of latent components. Our empirical results show that, by integrating metadata, MPF/MIPF significantly outperform the state-of-the-art PF models for sparse and large datasets. MIPF also effectively estimates the number of latent components.

1 Introduction

Recommender Systems (RSs) estimate ratings that users may give to corresponding items. One of the most popular classes of RSs is Collaborative Filtering (CF), by which ratings are estimated based on user's past behaviors and similar decisions made by other users. Based on such simple but effective intuition, CF-based recommender systems appear in many business systems e.g., Amazon. However, CF together with its central technique, Matrix Factorization (MF) [Koren *et al.*, 2009], faces many real-life challenges. First, it is not efficient to handle very large data, e.g., the Netflix data with millions of ratings, since classic MF requires intensive mathematical computation as discussed in [Mnih and Salakhutdinov, 2008; Gopalan *et al.*, 2015]. Second, real-life data is often very sparse, e.g., the Netflix data has 98.8% of the matrix entries missing; MF models fail to find the similar users since their way of computing similarities assumes that two users have rated at least some items in common. Lastly, choosing the suitable number of latent components (i.e., K) for large datasets in MF is a very expensive process as it requires testing many models.

To address the first MF weakness, Probabilistic-based MF models such as PMF [Mnih and Salakhutdinov, 2008] were proposed to handle large datasets. The underlying assumption of such models is that ratings are supposed to follow a specific distribution. However, they are still inefficient especially for sparse data, since they perform the computation on all data which usually consists of many missing (i.e., zero) ratings. The work on Poisson Factorization (PF) [Canny, 2004; Dunson and Herring, 2005; Gopalan *et al.*, 2015; Basbug and Engelhardt, 2016] shows that taking Poisson (with Gamma conjugate priors) as a distribution for non-missing values in sparse matrices leads to many advantages such as capturing non-negative values or only iterating over non-missing values. PF can partly address the second MF weakness on sparse data by ignoring the missing data in computation. Although computation only on non-missing data makes PF extremely fast for large and sparse datasets, it is inefficient when working with a column or row with very few observations (corresponding to a sparse item or user) due to poor priors.

Building on the non-IID recommender system view [Cao, 2016] and metadata-based coupling learning for MF models [Cao, 2015; Li *et al.*, 2015], this work introduces user and item metadata into PF and assume they follow the Gamma distribution as the priors of PF. A novel model: Metadata-dependent Poisson Factorization (MPF) is proposed to capture the couplings between user/item metadata and incorporate them into PF to capture both explicit and implicit relations in RS.

Further, to solve the third MF weakness, the BNP technique [Teh and Jordan, 2010] is integrated into MPF to form Metadata-dependent Infinite Poisson Factorization (MIPF). BNP allows for complex models to be learned without requiring much parameter tuning at the beginning. Consequently, we do not have to choose the number of latent components K in advance. MIPF automatically determines the number of latent components with an efficient variational inference [Jordan *et al.*, 1999] method to find the posterior.

Hence, our models make the following contributions.

- First, to the best of our knowledge, MPF is the first PF model that integrates user/item metadata into PF. MPF inherits the PF strength of modelling large and sparse datasets but provides richer priors for PF. Compared to other PF models, MPF shows its efficiency when dealing with the columns or rows with very few observations (i.e., sparse items or users).
- Further, the proposed MIPF is the first PF-based model that not only integrates the metadata but also can automatically choose the number of latent components.

Extensive experiments show that MPF outperforms the state-of-the-art models in the PF family for large datasets with sparse users/items; and MIPF retains the good performance as its finite version MPF while MPF has to fit many models before finding the best number of latent components.

2 The MPF/MIPF Models

Here we introduce the MPF and MIPF design and details.

2.1 Integrating Metadata into PF - The MPF Model

A recommendation problem usually consists of a rating matrix, and user and item information (called metadata here, e.g., the 'age', 'location' or 'career' of users and the 'genre' or 'year of release' of the movies in the Movielens dataset) [Cao, 2016]. Assume Y represents the rating matrix, in which each entry y_{ui} is the rating given by user u to item i, and an entry with 0 indicates no rating. HU and HI represent the user and item metadata respectively.

Poisson Factorization [Gopalan *et al.*, 2015] assumes the rating matrix Y follows the Poisson distribution and can be factorized to a vector of K latent preferences for each user, θ_{uk}, and a vector of K latent features, β_{ik}, for each item, where θ_{uk} and β_{ik} follow the Gamma distribution.

Building on PF, we further assume the Gamma distribution of each user's latent behavior, ξ_u, and each item's latent attractiveness, η_i. This hierarchical structure of Gamma-Gamma-Poisson allows us to capture the diversity of users and items. We capture the effects of user (item) metadata by defining the product of appearance of user (item) attribute's value in the metadata as the second parameter of Gamma distribution of user's latent behavior (item's latent attractiveness). We then apply the Gamma prior to the weight of each user attribute's value, hu_m, e.g., the 'New York' in 'location' attribute of a user, as in Eq. (1). The weight of user attribute hu_m only affects the behavior of a user ξ_u and further affects the preference similarity of users θ_{uk} if and only if $fu_{u,m} = 1$, as in Eq. (3). hu_m measures the degree of influence of each user attribute. For example, user 'location' may have less influence than 'age' in Movielens. The weight of an item attribute hi_n (e.g., 'genre' of a movie) is also given a Gamma distribution as in Eq. (2). 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 MPF is shown in Figure 1a and the generative process of MPF is as follows.

(1) For the m^{th} user attribute in the metadata, sample the weight:

$$hu_m \sim Gamma(\alpha_0, \alpha_1) \tag{1}$$

(2) For the n^{th} item attribute, sample the weight:

$$hi_n \sim Gamma(\gamma_0, \gamma_1) \tag{2}$$

(3) For each user u, sample latent behavior:

$$\xi_u \sim Gamma(a', \prod_{m=1}^{M} hu_m^{fu_{u,m}}) \tag{3}$$

(4) For each item i, sample latent attractiveness:

$$\eta_i \sim Gamma(c', \prod_{n=1}^{N} hi_n^{fi_{i,n}}) \tag{4}$$

(5) For each component k in the PF factorization:

(a) Sample user's latent preference:

$$\theta_{uk} \sim Gamma(a, \xi_u) \tag{5}$$

(b) Sample item's latent feature:

$$\beta_{ik} \sim Gamma(c, \eta_i) \tag{6}$$

(6) Sample rating:

$$y_{ui} \sim Poisson\left(\sum_k \theta_{uk}\beta_{ik}\right) \tag{7}$$

2.2 Taking the Infinite - The MIPF Model

The MIPF model extends MPF to handle infinite components by BNP. In MIPF, the user's latent preference θ_{uk} is constructed through the stick-breaking proportion v_{uk}, since stick-breaking process is efficient and widely used in many BNP models [Teh and Jordan, 2010; Liang *et al.*, 2007; Kurihara *et al.*, 2007; Gopalan *et al.*, 2014a]. After obtaining the number of latent components for users through the stick-breaking process, the distribution for items is given as in MPF.

The graphical model of MIPF is shown in Figure 1b and its generative process is below.

(1) For the m^{th} user attribute, sample the weight:

$$hu_m \sim Gamma(\alpha_0, \alpha_1) \tag{8}$$

(2) For the n^{th} item attribute, sample the weight:

$$hi_n \sim Gamma(\gamma_0, \gamma_1) \tag{9}$$

(3) For each user $u(= 1, \ldots, M)$:

(a) Draw the user's latent behavior:

$$\xi_u \sim Gamma(a', \prod_{m=1}^{M} hu_m^{fu_{u,m}}) \tag{10}$$

(b) For $k(= 1..\infty)$, draw stick-breaking proportion:

$$v_{uk} \sim Beta(1, a') \tag{11}$$

(c) For $k(= 1..\infty)$, set the user's latent preference:

$$\theta_{uk} = \xi_u . v_{uk} \prod_{l=1}^{k-1} (1 - v_{ul}) \quad (12)$$

(4) For each item $i(= 1...N)$:

(a) Draw the item's latent attractiveness:

$$\eta_i \sim Gamma(c', \prod_{n=1}^{N} hi_n^{fi_{i,n}}) \quad (13)$$

(b) For $k = (1...\infty)$, set the item's latent feature:

$$\beta_{ik} \sim Gamma(c, \eta_i) \quad (14)$$

(5) For $u(= 1...M)$ and $i(= 1...N)$, draw

$$y_{ui} \sim Poisson\bigg(\sum_{k=1}^{\infty} \theta_{uk}\beta_{ik} \bigg) \quad (15)$$

3 Inference

Applying MPF/MIPF to recommender systems has to solve the posterior inference problem. For this, the mean-field variational inference (VI) is incorporated into our models as it is more efficient for large-scale probabilistic models [Wainwright *et al.*, 2008] than other sampling approaches like Markov Chain Monte Carlo. With 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 divergence to the true posterior.

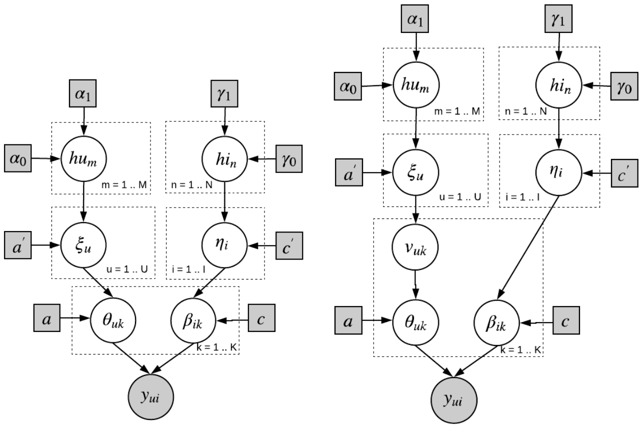
3.1 Variational Inference for MPF

Given the rating table Y together with the user/item metadata HU and HI, we compute the posterior distributions of the weight of user attribute in metadata hu_m, the weight of item attribute in metadata hi_n, the latent user preference θ_{uk}, the item's latent feature β_{ik}, the user's latent behavior ξ_u, and the item's latent attractiveness η_i. Taking the same approach as in [Gopalan *et al.*, 2015; Dunson and Herring, 2005; Zhou *et al.*, 2012; Gopalan *et al.*, 2014a], the rating y_{ui} is replaced with auxiliary latent variable $z_{ui,k} \sim Poisson(\theta_{uk}\beta_{ik})$. Due to the additive property of Poisson distribution, the rating y_{ui} is expressed as follows:

$$y_{ui} = \sum_k z_{ui,k} \quad (16)$$

Similar to [Gopalan *et al.*, 2015], the inference only considers $y_{ui} \neq 0$. The mean-field family assumes each distribution is independent of the others.

$$\begin{aligned} q(hu, hi, \theta, \beta, \xi, \eta, z) = & \prod_m q(hu_m|\zeta_m) \prod_n q(hi_n|\rho_n) \\ & \prod_{u,k} q(\theta_{uk}|\nu_{uk}) \prod_{i,k} q(\beta_{ik}|\mu_{ik}) \prod_u q(\xi_u|\kappa_u) \\ & \prod_i q(\eta_i|\tau_i) \prod_{u,i,k} q(z_{ui,k}|\phi_{ui,k}) \end{aligned} \quad (17)$$



(a) MPF (b) MIPF

Figure 1: The graphical models of MPF and MIPF.

We use the class of conditionally conjugate priors for hu_m, hi_n, θ_{uk}, β_{ik}, ξ_u, η_i and $z_{ui,k}$ to update the variational parameters $\{\zeta, \rho, \nu, \mu, \kappa, \tau, \phi\}$. For the Gamma distribution, we update both hyper-parameters: *shape* and *rate*.

(1) Update *shape* and *rate* of ζ_m:

$$\zeta_{m,0} = \alpha_0 + \aleph_m a' \quad (18)$$

$$\zeta_{m,1} = \alpha_1 + \sum_u \kappa_{u,0}/\kappa_{u,1} \quad (19)$$

where $\aleph_m$ is the number of users having attribute m.

(2) Update *shape* and *rate* of ρ_n:

$$\rho_{n,0} = \gamma_0 + \chi_n c' \quad (20)$$

$$\rho_{n,1} = \gamma_1 + \sum_i \tau_{i,0}/\tau_{i,1} \quad (21)$$

where χ_n is the number of items that have attribute n.

(3) Update *shape* and *rate* of κ_u:

$$\kappa_{u,0} = a' + Ka \quad (22)$$

$$\kappa_{u,1} = \prod_{m=1}^{M} (\zeta_{m,0}/\zeta_{m,1})^{fu_{u,m}} + \sum_k \frac{\nu_{uk,0}}{\nu_{uk,1}} \quad (23)$$

where K is the number of latent components.

(4) Update *shape* and *rate* of τ_i:

$$\tau_{i,0} = c' + Kc \quad (24)$$

$$\tau_{i,1} = \prod_{n=1}^{N} (\rho_{n,0}/\rho_{n,1})^{fi_{i,n}} + \sum_k \frac{\mu_{ik,0}}{\mu_{ik,1}} \quad (25)$$

(5) Update $\phi_{ui,k}$:

$$\begin{aligned} \phi_{ui,k} = & exp\{\Psi(\nu_{uk,0}) - log(\nu_{uk,1}) \\ & + \Psi(\mu_{ik,0}) - log(\mu_{ik,1})\} \end{aligned} \quad (26)$$

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

(6) Update *shape* and *rate* of ν_{uk}:

$$\nu_{uk,0} = a + \sum_i y_{ui}\phi_{ui,k} \quad (27)$$

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

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

$$\mu_{ik,0} = c + \sum_u y_{ui}\phi_{ui,k} \tag{29}$$

$$\mu_{ik,1} = \frac{\tau_{i,0}}{\tau_{i,1}} + \sum_u \frac{\nu_{uk,0}}{\nu_{uk,1}} \tag{30}$$

Owing to the limited space, we only give the deviation of integrating user/item metadata. The details of other deviations are similar to PF and are ignored here.

With the Gamma distribution in Eqs. (1) and (3),

$$p(hu_m|\alpha_0,\alpha_1) \propto hu_m^{\alpha_0-1} exp\{-\alpha_1 hu_m\} \tag{31}$$

$$p(\xi_u|a', hu_m) \propto \Big(\prod_{m=1}^{M} hu_m^{f_{u,m}a'}\Big) exp\{-(\prod_{m=1}^{M} hu_m^{f_{u,m}})\xi_u\} \tag{32}$$

The posterior probability of weight hu_m becomes:

$$p(hu_m|\alpha_0,\alpha_1,\xi_u) \propto p(hu_m|\alpha_0,\alpha_1)\prod_u p(\xi_u|a', hu_m) \\ \propto hu_m^{\alpha_0+\aleph_m a'-1} exp\{-(\alpha_1 + \sum_u \xi_u)hu_m\} \tag{33}$$

where $\aleph_m$ is the number of users having attribute m. The posterior Gamma distribution of hu_m is

$$hu_m \sim Gamma(\alpha_0 + \aleph_m a', \alpha_1 + \sum_u \xi_u) \tag{34}$$

hu_m is affected by $\aleph_m$ and the user's latent behavior (i.e., ξ_u). Similarly, the posterior distribution for the weight of item attribute, hi_n, is

$$hi_n \sim Gamma(\gamma_0 + \chi_n c', \gamma_1 + \sum_i \eta_i) \tag{35}$$

where χ_n is the number of items that have attribute n.

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 MPF is listed in Algorithm 1. We have to give the number of latent components at the beginning of the algorithm as in line 2.

3.2 Variational Inference for MIPF

Using the auxiliary variable as in Eq. (16), the completely factorized variational distribution in MIPF can be written as

$$q(hu, hi, v, \beta, \xi, \eta, z) = \prod_m q(hu_m|\zeta_m)\prod_n q(hi_n|\rho_n) \\ \prod_{k=1}^{\infty}\prod_u q(v_{uk}|\sigma_{uk})\prod_{k=1}^{\infty}\prod_i q(\beta_{ik}|\mu_{ik})\prod_u q(\xi_u|\kappa_u) \\ \prod_i q(\eta_i|\tau_i)\prod_{k=1}^{\infty}\prod_{u,i} q(z_{ui,k}|\phi_{ui,k}) \tag{36}$$

Algorithm 1 Variational Inference for MPF

```
1: Initialize the variational parameters $\{\zeta, \rho, \nu, \mu, \kappa, \tau, \phi\}$.
2: Set the number of components $K$.
3: Sample shape of user's latent behavior, and shape of item's latent attractiveness, as in Eqs. (22) and (24).
4: Sample shape of the weight of user's attribute (in metadata), and shape of the weight of item's attribute (in metadata), as in Eqs. (18) and (20).
5: repeat
6:   for each rating of user $u$ to item $i$ that $y_{ui} \neq 0$ do
7:     Update the multinominal as in Eq. (26).
8:   end for
9:   for each user do
10:    Update the latent preference as in Eqs. (27) and (28)
11:    Update rate of latent behavior as in Eq. (23).
12:    for each user attribute in metadata do
13:      Update rate of the weight as in Eq. (19)
14:    end for
15:  end for
16:  for each item do
17:    Update the latent feature as in Eqs. (29) and (30).
18:    Update rate of latent attractiveness as in Eq. (25).
19:    for each item attribute do
20:      Update rate of the weight as in Eq. (21).
21:    end for
22:  end for
23: until convergence
```

We update the variational parameters $\{\zeta, \rho, \mu, \tau\}$ similar to MPF. The new parameters needed to be updated are user's latent behavior κ, stick-breaking proportion σ and multinominal distribution ϕ. Again, owning to the limitation of space, we do not give the details of deviations of κ, σ and ϕ, which can be found at [Teh and Jordan, 2010; Liang *et al.*, 2007; Kurihara *et al.*, 2007; Gopalan *et al.*, 2014a].

(1) Update *shape* and *rate* of κ_u:

$$\kappa_{u,0} = a' + \prod_u y_{ui} \tag{37}$$

$$\kappa_{u,1} = \prod_{m=1}^{M}(\zeta_{m,0}/\zeta_{m,1})^{f_{u,m}} \\ +E\Big[\sum_{k=1}^{T} v_{uk}\Big(\prod_{j=1}^{k-1}(1-v_{uj})\Big)\sum_i \beta_{ik}\Big] \\ +\prod_{k=1}^{T}(1-\sigma_{uk})\sum_i E[\beta_{i(T+1)}] \tag{38}$$

where $E[\cdot]$ is the expectation with respect to the distribution q. T is the truncate level which is the upper bound of number of latent components K as described in [Kurihara *et al.*, 2007].

(2) Update the stick-breaking proportion σ_{uk} by solving the quadratic equation $A_{uk}\sigma_{uk}^2 + B_{uk}\sigma_{uk} + C_{uk} = 0$:

$$\sigma_{uk} = \frac{-B_{uk} \pm \sqrt{B_{uk}^2 - 4A_{uk}C_{uk}}}{2A_{uk}^2} \tag{39}$$

where the details of A_{uk}, B_{uk} and C_{uk} can be found in [Gopalan *et al.*, 2014a].

(3) Update the multinominal distribution $\phi_{ui,k}$

$$\phi_{ui,k} = \frac{exp\{R_{ui,k}\}}{\sum_{k=1}^{T} exp\{R_{ui,k}\} + \sum_{k=T+1}^{\infty} exp\{R_{ui,k}\}} \quad (40)$$

where

$$R_{ui,k} = E[log\theta_{uk}] + E[log\beta_{ik}] \quad (41)$$

and

$$\sum_{k=T+1}^{\infty} exp\{R_{ui,k}\} = \frac{exp\{R_{ui,T+1}\}}{1 - exp\{log(1 - v_{uT+1})\}} \quad (42)$$

The variational inference for MIPF is nearly similar to MPF with the exception in lines 2, 3, 7, 10 and 11. In line 2, instead of setting the number of latent components K, we set the truncate level T. In line 7, we update the multinominal as in Eq. (40). We update the stick-breaking proportion as in Eq. (39) instead of updating the user's preference as in line 10. In lines 3 and 11, we update the *shape* and *rate* of user's latent behavior as in Eqs. (37) and (38).

4 MPF/MIPF Properties and Related Work

We analyze the properties and performance of MPF/MIPF in the context of the related work.

(1) MPF/MIPF improve precision when working with large and sparse data by integrating user/item metadata. Given the vector of user's latent preferences θ_u and the vector of the item's latent features β_i, the probability of the rating by user u to item i, y_{ui}, based on Poisson distribution, is below.

$$p(y_{ui}|\theta_u, \beta_i) = \frac{(\theta_u^T \beta_i)^{y_{ui}} exp\{-\theta_u^T \beta_i\}}{y_{ui}!} \quad (43)$$

When $y_{ui} = 0$, it does not affect the probability. Similar to PF, it does not require optimization techniques to reduce the computational time as in the classical MF [Li *et al.*, 2015; Mairal *et al.*, 2010]. The probability only depends on θ_u and β_i. We provide richer priors by integrating user metadata to the user's behaviors, ξ_u, as in Eq. (44). The user's behaviors, ξ_u, in turn provide richer prior to the user's latent preferences θ_{uk}, as in Eq. (5).

$$\xi_u|\theta \sim Gamma(a' + Ka, \prod_{m=1}^{M} hu_m^{f_{u_{u,m}}} + \sum_k \theta_{uk}) \quad (44)$$

Similarly, we integrate item metadata to PF.

Different from the way of integrating document-word matrix into PF [Acharya *et al.*, 2015; Gopalan *et al.*, 2014b; Zhang and Wang, 2015; Hu *et al.*, 2016], MPF incorporates the user metadata, which includes more general attributes, e.g., categorical attributes, rather than just text. Recent work in [Zhao *et al.*, 2017; Fan *et al.*, 2017] also integrates such general attributes into probabilistic models for link prediction, but works only on small data due to the limitation of their inference.

Our models come with the variational inference, which has proved to be efficient for probabilistic models with a large amount of data. As MPF is built on the Gamma-Gamma-Gamma-Poisson distribution, it is fully Bayesian and conjugate. As discussed in [Ghahramani and Beal, 2001], we can easily build a variational algorithm for fully Bayesian and conjugate models. Although the VI method for MIPF is much more complicated since the distribution of MIPF is not in a closed form, there are some techniques to overcome this such as in [Liang *et al.*, 2007; Kurihara *et al.*, 2007; Gopalan *et al.*, 2014a].

(2) MIPF efficiently estimates the number of latent components. We use the same technique as in such BNP models as [Teh and Jordan, 2010; Liang *et al.*, 2007; Kurihara *et al.*, 2007; Gopalan *et al.*, 2014a] to estimate the number of latent components at running time. Different from the BNP MF, such as [Knowles and Ghahramani, 2011; Hoffman *et al.*, 2010], which requires to scan both missing and non-missing ratings, MIPF only computes on non-missing data. This makes our inference procedure extremely efficient for sparse matrices.

5 Experiments

5.1 Experimental Settings

Baseline Methods To the best of our knowledge, no existing methods have incorporated user/item metadata into PF and infinite PF. As [Gopalan *et al.*, 2015] shows that the hierarchical PF (HPF) outperforms baselines including basic PF, Nonnegative Matrix Factorization (NMF) [Berry *et al.*, 2007], Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003], and Probabilistic Matrix Factorization (PMF) [Mnih and Salakhutdinov, 2008], we thus here only compare our models with HPF, in addition to Bayesian Nonparametric PF (BNPPF) [Gopalan *et al.*, 2014a] and the latest PF: Hierarchical Compound PF (HCPF) [Basbug and Engelhardt, 2016].

Datasets MPF/MIPF are tested on four public datasets available with massive ratings and some metadata.

(1) Movielens100K [Harper and Konstan, 2016] contains $100,000$ ratings (from 1 to 5); user demographic: 'age', 'gender', 'occupation' and 'zip' ('age' is partitioned into the ranges: 1 : "$Under$ 18", 18 : "$18-24$", 25 : "$25-34$", 35 : "$35-44$", 45 : "$45-49$", 50 : "$50-55$", and 56 : "$56+$"); and item metadata: the 'genre','release date', and 'video release date' of movies.

(2) Movielens1M contains $1,000,209$ anonymous ratings and the same metadata as in Movielens100K.

(3) Movielens10M contains $10,000,054$ ratings with the metadata only containing the 'genre' of the movies.

(4) Book-Crossing [Ziegler *et al.*, 2005] contains $1,149,780$ ratings (from 1 to 10) with the user demographic: 'location' and 'age' ('age' is encoded in the same way as in Movielens100K) and the book information: 'book title', 'book author', 'year of publication', and 'publisher'.

Parameter Settings We set $a = c = a' = c' = 0.3$ in the same way as in HPF. The metadata hyper parameters α_0, α_1,

γ_0 and γ_1 are set to a small value: 0.1, so that the user/item attribute's weight automatically grows over time.

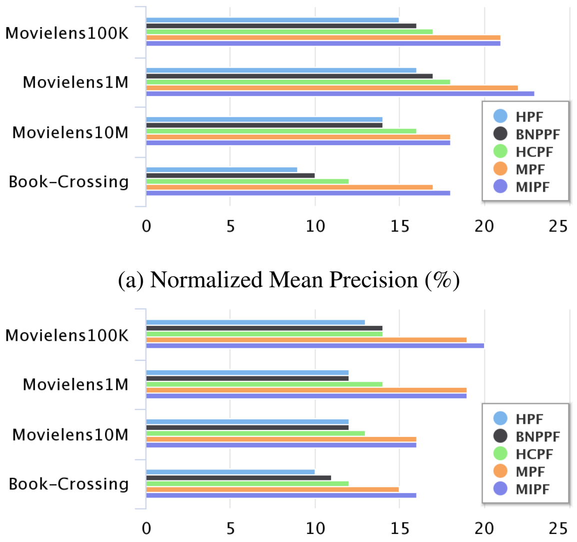
5.2 Result Evaluation

We evaluate MPF/MIPF in terms of their performance of addressing the three PF challenges (Q1, Q2 and Q3 below) w.r.t. the following metrics and convergence.

Evaluation Metrics We use 20% of the ratings for testing and 80% for training by random splitting. The top-N recommendations are obtained in training w.r.t. the highest prediction score. In testing, we compute the *precision-at-*N, which measures the fraction of relevant items in a user's top-N recommendations, and *recall-at-*N, which is the fraction of the testing items that 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.

Q1: How do MPF/MIPF significantly outperform other PF models? As shown in Figure 2, MPF/MIPF outperform HPF, BNPPF and HCPF in all datasets for normalized mean precision and normalized mean recall. The results on Movielens10M are not as good as on the others due to only one metadata attribute the 'genre' of the movies. MPF/MIPF make the most improvement (up to 9%) on Book-Crossing corresponding to the richest metadata available.



(a) Normalized Mean Precision (%)

(b) Normalized Mean Recall (%)

Figure 2: Top-20 recommendations compared with baselines.

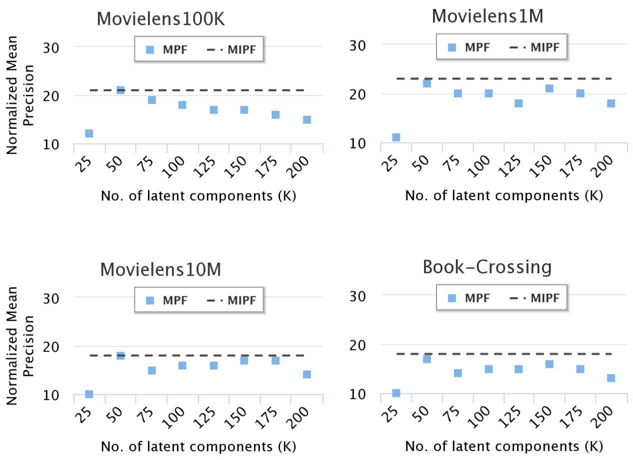


Figure 3: Performance of top-20 recommendations made by finite model MPF and infinite model MIPF.

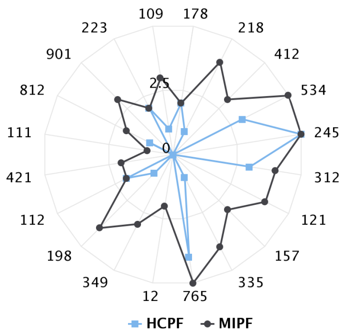


Figure 4: Example of MIPF in handling sparse items in comparison with HCPF.

Q2: How does MIPF effectively estimate the number of unbounded latent components? The infinite model MIPF is compared with the finite version MPF in Figure 3 in terms of normalized mean precision. The results for the normalized mean recall are consistent with normalized mean precision, omitted here due to the limitation of space. We run MPF w.r.t. K from 1 to 200 but MIPF for just once. For MIPF, we set the truncate level T to 200. We can see that MIPF can always achieve as good as the best results of MPF. This shows MIPF is very efficient in selecting the suitable number of latent components K.

Q3: How do MPF/MIPF deal with sparse items/users? In this experiment, we choose 20 users who have the highest number of precisely recommended items. We calculate the sparsity within each item (i.e., the fraction of the number of users who gave ratings for that item in the total number of users). Figure 4 shows the percentage of number of items that have sparsity less than 1% in the total recommended items for each user for MIPF and HCPF. It shows MIPF recommends more sparse items than HCPF as the result of integrating metadata. The results for sparse users are similar to sparse items, omitted here for the limitation of space.

6 Conclusions

Two novel and efficient Poisson factorization models MPF and MIPF are proposed for sparse and large-scale recommendation. They are the first to effectively integrate user/item metadata into PF, and MIPF can effectively estimate the number of latent components in just one run. We are developing even more efficient inference for MIPF to handle increasingly bigger data.

References

[Acharya *et al.*, 2015] Ayan Acharya, Dean Teffer, Jette Henderson, Marcus Tyler, Mingyuan Zhou, and Joydeep Ghosh. Gamma process Poisson factorization for joint modeling of network and documents. In *ECML PKDD*, pages 283–299. Springer, 2015.

[Basbug and Engelhardt, 2016] Mehmet Basbug and Barbara Engelhardt. Hierarchical compound poisson factorization. In *ICML*, pages 1795–1803, 2016.

[Berry *et al.*, 2007] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 3(Jan):993–1022, 2003.

[Canny, 2004] John Canny. Gap: a factor model for discrete data. In *SIGIR*, pages 122–129. ACM, 2004.

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

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

[Dunson and Herring, 2005] David B Dunson and Amy H Herring. Bayesian latent variable models for mixed discrete outcomes. *Biostatistics*, 6(1):11–25, 2005.

[Fan *et al.*, 2017] Xuhui Fan, Richard Yi Da Xu, Longbing Cao, and Yin Song. Learning nonparametric relational models by conjugately incorporating node information in a network. *IEEE Transactions on Cybernetics*, 47(3):589–599, 2017.

[Ghahramani and Beal, 2001] Zoubin Ghahramani and Matthew J Beal. Propagation algorithms for variational bayesian learning. In *NIPS*, pages 507–513, 2001.

[Gopalan *et al.*, 2014a] Prem Gopalan, Francisco J Ruiz, Rajesh Ranganath, and David Blei. Bayesian nonparametric poisson factorization for recommendation systems. In *AISTATS*, pages 275–283, 2014.

[Gopalan *et al.*, 2014b] Prem K Gopalan, Laurent Charlin, and David Blei. Content-based recommendations with Poisson Factorization. In *NIPS*, pages 3176–3184, 2014.

[Gopalan *et al.*, 2015] Prem Gopalan, Jake M Hofman, and David M Blei. Scalable Recommendation with Hierarchical Poisson Factorization. In *UAI*, pages 326–335, 2015.

[Harper and Konstan, 2016] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM TiiS*, 5(4):19, 2016.

[Hoffman *et al.*, 2010] Matthew D Hoffman, David M Blei, and Perry R Cook. Bayesian nonparametric matrix factorization for recorded music. In *ICML*, pages 439–446, 2010.

[Hu *et al.*, 2016] Changwei Hu, Piyush Rai, and Lawrence Carin. Topic-based embeddings for learning from large knowledge graphs. In *AISTATS*, pages 1133–1141, 2016.

[Jordan *et al.*, 1999] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[Knowles and Ghahramani, 2011] David Knowles and Zoubin Ghahramani. Nonparametric bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, pages 1534–1552, 2011.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[Kurihara *et al.*, 2007] Kenichi Kurihara, Max Welling, and Nikos Vlassis. Accelerated variational dirichlet process mixtures. In *NIPS*, pages 761–768, 2007.

[Li *et al.*, 2015] Fangfang Li, Guandong Xu, and Longbing Cao. Coupled matrix factorization within non-iid context. In *PAKDD (2)*, pages 707–719, 2015.

[Liang *et al.*, 2007] Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. The infinite pcfg using hierarchical dirichlet processes. In *EMNLP-CoNLL*, pages 688–697, 2007.

[Mairal *et al.*, 2010] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11(Jan):19–60, 2010.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2008.

[Teh and Jordan, 2010] Yee Whye Teh and Michael I Jordan. Hierarchical bayesian nonparametric models with applications. *Bayesian nonparametrics*, 1, 2010.

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

[Zhang and Wang, 2015] Wei Zhang and Jianyong Wang. A collective Bayesian Poisson Factorization model for cold-start local event recommendation. In *SIGKDD*, pages 1455–1464. ACM, 2015.

[Zhao *et al.*, 2017] He Zhao, Lan Du, and Wray Buntine. Leveraging node attributes for incomplete relational data. *ICML*, 2017.

[Zhou *et al.*, 2012] Mingyuan Zhou, Lauren Hannah, David B Dunson, and Lawrence Carin. Beta-negative binomial process and poisson factor analysis. In *AISTATS*, pages 1462–1471, 2012.

[Ziegler *et al.*, 2005] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *WWW*, pages 22–32. ACM, 2005.