Bayesian Nonparametric Poisson Factorization for Recommendation Systems

Prem Gopalan
Princeton University

Francisco J. R. Ruiz
University Carlos III in Madrid

Rajesh Ranganath
Princeton University

David M. Blei
Princeton University

Abstract

We develop a Bayesian nonparametric Poisson factorization model for recommendation systems. Poisson factorization implicitly models each user's limited budget of attention (or money) that allows consumption of only a small subset of the available items. In our Bayesian nonparametric variant, the number of latent components is theoretically unbounded and effectively estimated when computing a posterior with observed user behavior data. To approximate the posterior, we develop an efficient variational inference algorithm. It adapts the dimensionality of the latent components to the data, only requires iteration over the user/item pairs that have been rated, and has computational complexity on the same order as for a parametric model with fixed dimensionality. We studied our model and algorithm with large real-world data sets of user-movie preferences. Our model eases the computational burden of searching for the number of latent components and gives better predictive performance than its parametric counterpart.

1 Introduction

Recommendation systems try to predict which items a user will like based on his or her history of purchases or ratings. One of the most important classes of recommendation methods is collaborative filtering, where we extract information from the population of users to form specific recommendations for each one. Specifically, collaborative filtering methods analyze a collection of user/item purchase data, try to find recurring patterns of item preferences among the users, and then exploit those patterns for each user to recommend items that he or she has not yet purchased.

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

In this paper, we extend Poisson factorization (Canny, 2004; Dunson & Herring, 2005; Cemgil, 2009; Gopalan et al., 2013) for recommendation. Poisson factorization (PF) models each user/item observation with a Poisson whose rate is the dot product of the latent, non-negative user and item weights. The main computational problem is posterior inference, i.e., discovering the latent user and item weights given an observed user/item matrix.

This general procedure is common to many variants of matrix factorization. However, PF enjoys advantages over classical methods on a variety of data sets, including those with implicit feedback (a binary matrix indicating which items users consumed) and those with explicit feedback (a matrix of integer ratings). Classical matrix factorization, which corresponds to a Gaussian model of the data (Salakhutdinov & Mnih, 2008), requires complex methods for downweighting the effect of zeros in the implicit setting (Hu, Koren, & Volinsky, 2008; Gantner, Drumond, Freudenthaler, & Schmidt-Thieme, 2012; Dror, Koenigstein, Koren, & Weimer, 2012), while PF treats all zeros as observations, implicitly capturing their effect as a limitation on user resources (Gopalan et al., 2013).

The second advantage of PF algorithms is that they need only iterate over the viewed items in the observed matrix of user behavior, i.e., the non-zero elements, and this is true even for implicit or "positive only" data sets. (This follows from the mathematical form of the Poisson distribution.) Thus, Poisson factorization takes advantage of the natural sparsity of user behavior data and scales to massive real-world data. In contrast, classical matrix factorization must iterate over both positive and negative examples in the implicit setting. Thus it cannot take advantage of data sparsity, which makes computation difficult for even modestly sized problems in the implicit setting.

A limitation of matrix factorization is model selection, i.e., choosing the number of components with which to model the data. The typical approach in matrix factorization, Poisson or Gaussian, is to determine the number of components by predictive performance on a held-out set of ratings (Salakhutdinov & Mnih, 2008; Gopalan et al., 2013). But this can be prohibitively expensive with large data sets because it requires fitting many models.

To this end, we develop a Bayesian nonparametric (BNP) Poisson factorization model. Our model is based on the weights from a collection of Gamma processes, one for each user, which share an infinite collection of atoms. Each atom represents a preference pattern of items, such as action movies or comedies. Through its posterior distribution, our model adapts the dimensionality of the latent representations, learning the preference patterns (and their number) that best describe the users.

As for most complex Bayesian models, the main computational challenge is posterior inference, where we approximate the posterior distribution of the latent user/item structure given a data set of user/item ratings. We develop an efficient algorithm based on variational inference, an approach that finds an approximate posterior distribution via optimization (Jordan et al., 1999). Our method simultaneously finds both the latent components and the latent dimensionality, easily handles large data sets, and takes time roughly equal to one fit of the model with fixed dimension.

Thus, the contributions of our paper are: (a) a new Bayesian nonparametric model for Poisson factorization, (b) a scalable variational inference algorithm with nested variational families (Kurihara et al., 2006), and (c) a thorough study of BNPPF on large scale movie recommendation problems. On two large real-world data sets—10M movie ratings from MovieLens and 100M movie ratings from Netflix—we found that Bayesian nonparametric PF outperformed (or at least performed as well as) its parametric counterpart.

There has been significant research on Bayesian nonparametric factor models. Griffiths and Ghahramani (2011) introduced the IBP, and developed a Gaussian BNP factor model with binary weights. Knowles and Ghahramani (2011) later extended this work to non-binary weights. Other extensions include (Hoffman et al., 2010) and (Porteous et al., 2008). Situated within this literature, our model is a BNP factor model that assumes non-negative weights and sparse observations. As we will show below, unlike previous algorithms, our approach takes advantage of the sparsity and non-negativity to scale to very large data sets. We analyze data that cannot be handled by this previous research.

Closer to our model is the work of Titsias (2007), Zhou et al. (2012), and Broderick et al. (2013). Titsias (2007) derives a BNP factor model, the infinite Gamma-Poisson feature model. The Gamma process can be shown to be the De Finetti mixing distribution for this model (Thibaux, 2008), with the latent counts being drawn from a Poisson process. Our model does not have an underlying latent discrete stochastic process. Zhou et al. (2012) generalize (Titsias, 2007) and extend the infinite prior to a Beta-Gamma-Gamma-Poisson hierarchical structure. Our model is simpler than (Zhou et al., 2012) because it is not hierarchical, and our model choices afford a scalable variational inference algorithm to tackle the kinds of problems that we are trying to solve. Titsias (2007) uses an MCMC algorithm that does not scale and Zhou et al. (2012) further do inference with a truncated model (which also does not scale). Finally, Broderick et al. (2013) give a more general class of hierarchical models than (Zhou et al., 2012) but, again, only develop MCMC algorithms. Our model is simpler than and complements (Broderick et al., 2013)—Poisson factorization does not immediately fall out of their framework—and, again, affords scalable inference algorithms.

Finally, we remark that although we focus on recommendation systems, our BNP model can be used in a broader range of applications requiring matrix factorization, such as topic modeling (Canny, 2004) or community detection (Ball et al., 2011).

2 Bayesian Nonparametric Poisson Factorization

We develop a statistical model of user/item rating matrices. Our data are observations y_{ui}, which contains the rating that user u gave to item i, or zero if no rating was given. The data can be based on "implicit feedback", where y_{ui} is one if the user consumed it and zero otherwise. User behavior data is typically sparse, i.e., most of the ratings are zero.

In Poisson factorization for recommendation (Gopalan et al., 2013), each user and each item is associated with a K-dimensional latent vector of positive weights. Let $\boldsymbol{\theta}_u = [\theta_{u1}, \ldots, \theta_{uK}]^\top$ be the weight vector of user u, and $\boldsymbol{\beta}_i = [\beta_{i1}, \ldots, \beta_{iK}]^\top$ be the weight vector of item i. The weights are given Gamma priors and each observation y_{ui} is modeled by a Poisson distribution parameterized by the inner product of the user's and item's weights,

$$\beta_{ik} \sim \text{Gamma}(a, b), \tag{1}$$

$$\theta_{uk} \sim \text{Gamma}(c, d), \tag{2}$$

$$y_{ui} \sim \text{Poisson}(\boldsymbol{\theta}_u^\top \boldsymbol{\beta}_i). \tag{3}$$

The number of components is fixed. Throughout the paper we parameterize the Gamma with its shape and rate. Gopalan et al. (2013) fit the model with different values of K and select the one that gives best performance on a held-out set of ratings.

Choosing the value of K is a nuisance because it is expensive to fit many models on large data sets. We want a model with support over arbitrary K such that the posterior distribution captures the effective latent dimensionality of our data. The desirable properties for such a model are: (i) The user weights $\boldsymbol{\theta}_u$ and item weights $\boldsymbol{\beta}_i$ must be infinite-dimensional non-negative vectors, (ii) the expected dot product $\boldsymbol{\theta}_u^\top \boldsymbol{\beta}_i$ of any pair must be finite, and (iii) the item weights $\boldsymbol{\beta}_i$ have to be shared among all users.

We place independent and identical Gamma priors over each element of the infinite-dimensional vector $\boldsymbol{\beta}_i$, following Eq. 1. Since all elements β_{ik} are iid, to satisfy condition (ii), we require the convergence of $\sum_{k=1}^{\infty} \mathbb{E}[\theta_{uk}]$. A natural way to construct a summable infinite set of positive weights is to use a Gamma process (Ferguson, 1973) for each user weight vector $\boldsymbol{\theta}_u$. A Gamma process, $\mathrm{GP}(c, H)$, has two parameters: c the rate parameter and H the base measure. A draw from a Gamma process is an atomic random measure with finite total mass when H is finite. This means that a draw from a Gamma process gives an infinite collection of positive-valued random weights (the atom weights) whose summation is almost surely finite. Thus, we make θ_{uk} the weights of a draw from a Gamma process.

To ensure the items are shared among all users and satisfy condition (iii), we need to match the item weights to the user weights. Notice that the item weights are indexed by the natural numbers. Thus, to match the user weights to the item weights, we need an ordering of the user weights. We use a size-biased ordering. The size-biased ordering promotes sharing by penalizing higher index components. We obtain per-user Gamma process weights, in size-biased order, by scaling the stick-breaking construction of the Dirichlet process with a Gamma random variable. Miller (2011) states this construction for Gamma processes with unit scale. In general, scaling the sticks from a Dirichlet process by a draw from $\mathrm{Gamma}(\alpha, c)$ yields a draw from a Gamma process with parameters c and H, where $H(\Omega) = \alpha$ (Zhou & Carin, 2013). While α governs the sparsity of the user weights, both α and c influence the rating budget available to the user.

Using the scaled sticks for $\boldsymbol{\theta}_u$, the generative process for the Bayesian nonparametric Poisson factorization model with N users and M items is as follows:

1. For each user $u = 1, \ldots, N$:
   (a) Draw $s_u \sim \mathrm{Gamma}(\alpha, c)$.
   (b) Draw $v_{uk} \sim \mathrm{Beta}(1, \alpha)$, for $k = 1, \ldots, \infty$.
   (c) Set $\theta_{uk} = s_u \cdot v_{uk} \prod_{i=1}^{k-1}(1 - v_{ui})$, for $k = 1, \ldots, \infty$.
2. Draw $\beta_{ik} \sim \mathrm{Gamma}(a, b)$, for $k = 1, \ldots, \infty$, $i = 1, \ldots, M$.
3. Draw $y_{ui} \sim \mathrm{Poisson}(\sum_{k=1}^{\infty} \theta_{uk}\beta_{ik})$, for $u = 1, \ldots, N$, $i = 1, \ldots, M$.

Unlike draws from a standard Gamma process, these atoms are shared across users. In this way, our model is similar to a hierarchical Gamma process (Çinlar, 1975). However, in a hierarchical Gamma process, the atoms are shared across users through the common base measure. In contrast, the atoms in our model are shared due to the size-biased ordering. Intuitively, components with different indices are unlikely to be similar due to the penalty levied by the size-biased ordering. This method of sharing atoms can be leveraged in other hierarchical BNP models like the hierarchical Dirichlet process (Teh & Jordan, 2010).

Note that, unlike BNP Gaussian matrix factorization (Knowles & Ghahramani, 2011), the generative process provides a sparse observation matrix. The probability of y_{ui} being zero can be lower bounded as

$$p(y_{ui} = 0) \geq \exp\left(-\frac{a\alpha}{bc}\right) \tag{4}$$

and, therefore, the expected number of zeros in the observation matrix is lower bounded by $NM \exp(-\frac{a\alpha}{bc})$ (see the Supplementary Material for the proof).

3 Inference

In this section, we derive a scalable mean-field inference algorithm to deal with large data sets. Variational inference provides an alternative to MCMC methods as a general source of approximation methods for inference in large-scale probabilistic models (Jordan et al., 1999). Variational inference algorithms are in general computationally less expensive compared to MCMC methods and do not suffer from limitations involving mixing of the Markov chains, although they involve solving a non-convex optimization problem.

Variational inference algorithms approximate the posterior by defining a parametrized family of distributions over the hidden variables and then fitting the parameters to find a distribution that is close to the posterior in terms of Kullback-Leibler divergence.

For convenience, and similarly to previous works (Dunson & Herring, 2005; Zhou et al., 2012; Gopalan et al., 2013), we introduce for each user-item pair the auxiliary latent variables $z_{ui,k}$, such that $z_{ui,k} \sim \text{Poisson}(\theta_{uk}\beta_{ik})$. Due to the additive property of Poisson random variables, each observation y_{ui} can be expressed as

$$y_{ui} = \sum_{k=1}^{\infty} z_{ui,k}. \tag{5}$$

Thus, the variables $z_{ui,k}$ preserve the marginal Poisson distribution of the observation y_{ui}. Note that when $y_{ui} = 0$, the posterior distribution of $z_{ui,k}$ will place all its mass on $z_{ui,k} = 0$. Consequently, our inference procedure needs to only consider $z_{ui,k}$ for those user-item pairs such that $y_{ui} > 0$. This is not the case for BNP Gaussian MF (Knowles & Ghahramani, 2011) and makes our inference procedure extremely efficient for sparse user/item data.

Using the auxiliary variables, and introducing the notation $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i\}$, $\boldsymbol{s} = \{s_u\}$, $\boldsymbol{v} = \{v_{uk}\}$ and $\boldsymbol{z} = \{z_{ui,k}\}$, the joint distribution over the hidden variables can be written as

$$\begin{aligned} p(\boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{s}, \boldsymbol{v} | \alpha, c, a, b) &= \prod_{u=1}^{N} p(s_u|\alpha, c) \prod_{k=1}^{\infty}\prod_{u=1}^{N} p(v_{uk}|\alpha) \\ \times \prod_{k=1}^{\infty}\prod_{i=1}^{M} p(\beta_{ik}|a, b) &\prod_{k=1}^{\infty}\prod_{u=1}^{N}\prod_{i=1}^{M} p(z_{ui,k}|\theta_{uk}, \beta_{ik}), \end{aligned} \tag{6}$$

and the observations are generated following Eq. 5.

The mean-field family considers the latent variables to be independent of each other, yielding the completely factorized variational distribution:

$$\begin{aligned} q(\boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{s}, \boldsymbol{v}) = \prod_{u=1}^{N} q(s_u) \prod_{k=1}^{\infty}\prod_{u=1}^{N} q(v_{uk}) \prod_{k=1}^{\infty}\prod_{i=1}^{M} q(\beta_{ik}) \\ \times \prod_{u=1}^{N}\prod_{i=1}^{M} q(\boldsymbol{z}_{ui}), \end{aligned} \tag{7}$$

in which we denote by $\boldsymbol{z}_{ui}$ the vector with the infinite collection of variables $z_{ui,k}$ for user u and item i.

Typical mean field methods optimize the KL divergence by coordinate ascent, iteratively optimizing each parameter while holding the others fixed. These update are easy to derive for conditionally conjugate variables, i.e., variables whose complete conditional is in the exponential family (Ghahramani & Beal, 2001). (The complete conditional is the conditional distribution of a latent variable given the observations and all other latent variables). This is the case for most of the variables in our model and for these variables we set the form of their variational distributions to be the same as their complete conditionals.

The exceptions are the stick proportions. These variables are not conditionally conjugate because the Beta prior over the stick proportions v_{uk} is not conjugate to the Poisson likelihood. Rather, the complete conditional for the stick proportions v_{uk} is a truncated Gamma distribution. Letting the variational distribution be in the prior or the complete conditional family results in coordinate updates that do not have a closed form. Therefore, we resort to a degenerate delta distribution for $q(v_{uk})$, i.e., $q(v_{uk}) = \delta_{\tau_{uk}}(v_{uk})$, an alternative that is widely used in the BNP literature (Liang et al., 2007; Bryant & Sudderth, 2012).[1]

Finally, we handle the infinite collection of variational factors in Eq. 7 by adapting the technique of Kurihara et al. (2006), in which the variational families are nested over a truncation level T. We allow for an infinite number of components for the variational distribution, but we tie the variational distribution after level T to the prior. Specifically, $q(v_{uk})$ and $q(\beta_{ik})$ are set to the prior for $k \geq T+1$.

With these considerations in mind, the forms of the variational distributions in Eq. 7 are as follows:

$$\begin{aligned} &q(s_u) = \text{Gamma}(s_u|\gamma_{u,0}, \gamma_{u,1}), \\ &q(v_{uk}) = \begin{cases} \delta_{\tau_{uk}}(v_{uk}), & \text{for } k \leq T, \\ p(v_{uk}), & \text{for } k \geq T+1, \end{cases} \\ &q(\beta_{ik}) = \begin{cases} \text{Gamma}(\beta_{ik}|\lambda_{ik,0}, \lambda_{ik,1}), & \text{for } k \leq T, \\ p(\beta_{ik}), & \text{for } k \geq T+1, \end{cases} \\ &q(\boldsymbol{z}_{ui}) = \text{Mult}(\boldsymbol{z}_{ui}|y_{ui}, \boldsymbol{\phi}_{ui}). \end{aligned} \tag{8}$$

We now describe the specific updates for each variational parameter. Figure 1 gives the algorithm.

1. The update equations for the *user scaling parameters* $\gamma_{u,0}$ and $\gamma_{u,1}$ are given by

$$\gamma_{u,0} = \alpha + \sum_{i=1}^{M} y_{ui}, \tag{9}$$

$$\gamma_{u,1} = c + \mathbb{E}\left[\sum_{k=1}^{\infty} v_{uk} \left(\prod_{j=1}^{k-1}(1 - v_{uj})\right) \sum_{i=1}^{M} \beta_{ik}\right], \tag{10}$$

where $\mathbb{E}[\cdot]$ denotes expectation with respect to the distribution q. The infinite sum in the update equation for $\gamma_{u,1}$ can be split into the sum of the

[1]In variational inference, minimizing the KL divergence is equivalent to maximizing an objective function called ELBO (evidence lower bound). When the support of the variational distribution and the true posterior do not coincide, maximizing the ELBO is not equivalent to minimizing the KL divergence. In our case, $q(v_{uk})$ is a degenerate delta function and, therefore, its support is not the whole interval $[0, 1]$. The resulting algorithm that maximizes the ELBO can be understood instead as a variational expectation maximization algorithm (Beal & Ghahramani, 2003).

Given a truncation level T, for all users and items, initialize the user scaling parameters $\{\gamma_{u,0}, \gamma_{u,1}\}$ and the item parameters $\{\lambda_{ik,0}, \lambda_{ik,1}\}$ to the prior with a small random offset. Initialize the stick proportions τ_{uk}, where $k \leq T$, randomly.

Repeat until convergence:

1. For each user/item pair such that $y_{ui} > 0$,
   - Update the multinomial parameter $\boldsymbol{\phi}_{ui}$ using Eq. 16.
2. For each user,
   - Update the user scaling parameters $\gamma_{u,0}$ and $\gamma_{u,1}$ using Eq. 9 and Eq. 10.
   - Update the user stick proportions τ_{uk} for all $k \leq T$ using Eq. 12.
3. For each item,
   - Update the item weight parameters $\lambda_{ik,0}$ and $\lambda_{ik,1}$ using Eq. 13 and Eq. 14 for all $k \leq T$.

Figure 1: Batch variational inference for the Bayesian nonparametric Poisson factorization model. Each iteration only needs to consider the non-zero elements of the user/item matrix.

terms for $k \leq T$ and the sum of the terms for $k \geq T+1$. The first sum is straightforward to compute, but the second one involves infinitely many terms. However, it results in a convergent geometric series whose value is given by

$$\mathbb{E}\left[\sum_{k=T+1}^{\infty} v_{uk} Y_{uT} \left(\prod_{j=T+1}^{k-1}(1-v_{uj})\right)\sum_{i=1}^{M}\beta_{ik}\right] = Y_{uT} D, \qquad (11)$$

where $Y_{uT} = \prod_{k=1}^{T}(1-\tau_{uk})$ and $D = \sum_{i=1}^{M} \mathbb{E}\left[\beta_{i(T+1)}\right] = Ma/b$.

2. The update equations for the *stick proportions* τ_{uk} can be obtained by taking the derivative of the objective function with respect to τ_{uk} and setting it equal to zero. This yields the quadratic equation $A_{uk}\tau_{uk}^2 + B_{uk}\tau_{uk} + C_{uk} = 0$, where the coefficients A_{uk}, B_{uk} and C_{uk} are provided in the Supplementary Material. Solving for τ_{uk}, we get

$$\tau_{uk} = \frac{-B_{uk} \pm \sqrt{B_{uk}^2 - 4A_{uk}C_{uk}}}{2A_{uk}}, \qquad (12)$$

and we discard the solution that is not in $[0,1]$. Note that we require $\alpha > 1$ for the variational objective to be a concave function of τ_{uk}.

3. For the *item weights*, the equations for the variational parameters $\lambda_{ik,0}$ and $\lambda_{ik,1}$ are straightforward due to the conditional conjugacy of the distributions involved:

$$\lambda_{ik,0} = a + \sum_{u=1}^{N} y_{ui}\phi_{ui,k}, \qquad (13)$$

$$\lambda_{ik,1} = b + \sum_{u=1}^{N} \mathbb{E}\left[s_u v_{uk} \prod_{j=1}^{k-1}(1-v_{uj})\right]. \qquad (14)$$

4. Exploiting the Gamma-Poisson conjugacy, we know that the optimal $q(\boldsymbol{z}_{ui})$ can be parameterized by an infinite-dimensional vector $\boldsymbol{\phi}_{ui}$ whose components take the form

$$\phi_{ui,k} \propto \exp\{\mathbb{E}\left[\log\theta_{uk}\right] + \mathbb{E}\left[\log\beta_{ik}\right]\}. \qquad (15)$$

Let $R_{ui,k} = \mathbb{E}\left[\log\theta_{uk}\right] + \mathbb{E}\left[\log\beta_{ik}\right]$. Then,

$$\phi_{ui,k} = \frac{\exp\{R_{ui,k}\}}{\sum_{k=1}^{\infty}\exp\{R_{ui,k}\}}. \qquad (16)$$

Similarly to the derivation by Kurihara et al. (2006), we are left with computing a normalizer that is an infinite sum. The summation up to the truncation level T is straightforward, and thus we focus on computing $\sum_{k=T+1}^{\infty}\exp\{R_{ui,k}\}$. It is also a convergent geometric series, which can be computed as

$$\sum_{k=T+1}^{\infty}\exp\{R_{ui,k}\} = \frac{\exp\{R_{ui,T+1}\}}{1-\exp\left\{\mathbb{E}\left[\log(1-v_{u(T+1)})\right]\right\}}. \qquad (17)$$

We have described our batch variational inference algorithm for the BNPPF. We emphasize that the algorithm needs to only iterate over the nonzero elements. For a dataset with N users, M items and r ratings, the algorithm in Figure 1 has a computational complexity of $O(T^2N + TM + Tr)$. The dominant cost is the iteration over ratings captured by the $O(Tr)$ term, which equals the cost for the finite PF model with a fixed number of components $K = T$ (Gopalan et al., 2013). This allows us, in the next section, to analyze very large user/item data sets.

4 Empirical Study

In this section, we compare the nonparametric Poisson Factorization model (BNPPF) and the finite Poisson factorization model (Gopalan et al., 2013) on three

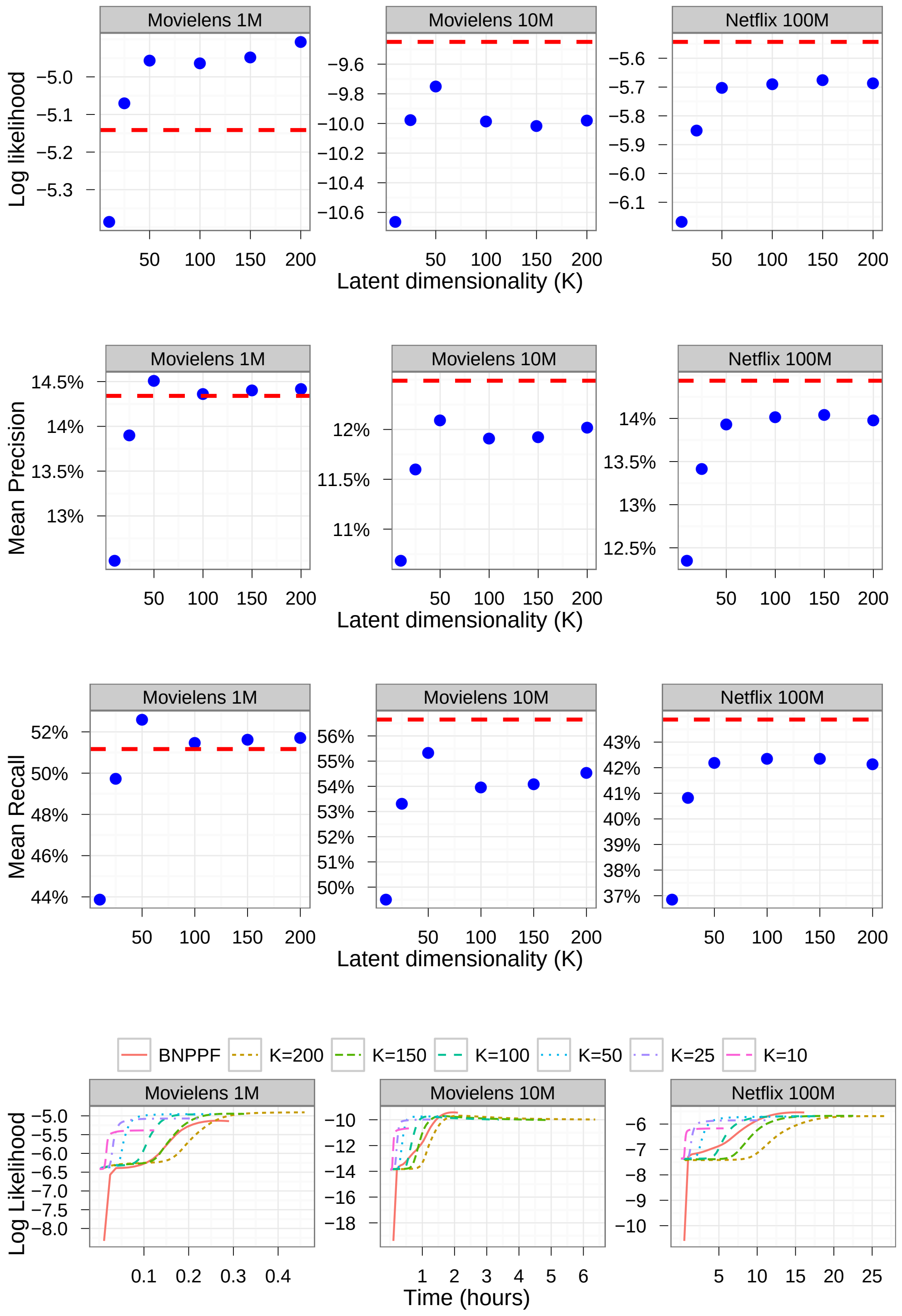

Figure 2: Rows 1-3: Generalization performance on the MovieLens and the Netflix data sets. The data sets vary in size from 1 million ratings to 100 million ratings. Points indicate the finite model predictive performance with varying dimensionality K, while the horizontal line indicates the BNPPF model performance. The BNPPF model demonstrates better predictive log likelihood for the two largest data sets and is at least as good as the finite models in terms of mean precision and recall. Row 4: Predictive log likelihood as a function of run-time for all the considered models. The BNPPF model is as fast as a single run of the finite model with $K = T$.

different data sets. Our goal is to demonstrate that the BNPPF variational inference algorithm can avoid model selection, while yielding better or similar performance than the variational inference algorithm for its finite counterpart. On large data sets, model selection involves fitting the finite model on a wide range of the latent dimensionality K, making it computationally intractable. We also show that our inference algorithm scales, and that it runs in roughly the same time needed for a single run of the finite model. We do not compare to Gaussian BNP matrix factorization algorithms which require iteration over all elements of the user/item matrix. We note that the finite algorithm outperforms Gaussian MF (Gopalan et al., 2013).

We implemented the algorithm of Figure 1 in $4,000$ lines of C++ code. The input to this algorithm is the user/item matrix. The output are the parameters for the approximate posterior distribution over the user and item weights.[2]

Data sets. We study the predictive performance and runtime of the BNPPF model on these data sets:

–MovieLens dataset (Herlocker et al., 1999), which contains 1 million (MovieLens1M) ratings of movies provided by users, with $6,040$ users and $3,980$ movies. The ratings range from 0 (no rating) to 5 stars.

–MovieLens dataset with 10 million (MovieLens10M) ratings of movies, $71,567$ users and $10,681$ movies. Ratings are made on a 5-star scale, with half-star increments. We multiply times 2 to get a scale from 0 to 10 "half stars".

–Netflix dataset (Koren et al., 2009), which is similar to MovieLens1M dataset but significantly larger. It contains 100 million ratings, with $480,000$ users and $17,770$ movies. Unlike MovieLens, the Netflix data is highly skewed: some users rate more than $10,000$ movies, while others rate less than 5.

Metrics. As figures of merit, we use the quantities below, measured over a held-out test set which is not observed during training. For each dataset, the test set consists of randomly selected ratings which make up 20% of the total number of ratings. This test set consists of items that the users have consumed. During training, these test set observations are treated as zeros.

–Predictive log likelihood (or mean held-out log likelihood). We approximate the probability that a user consumed an item using the variational approximations to the posterior expectations of the hidden variables. For the BNPPF model, we compute the expectation $\mathbb{E}\left[\sum_{k=1}^{\infty}\theta_{uk}\beta_{ik}\right]$ exactly using a convergent geometric series as in the updates in Eq. 11. We use these expectations to compute the average predictive log likelihood of the held-out ratings.

–Mean precision and recall. Once the posterior is fit, we use the BNPPF to recommend items to users by predicting which of the unconsumed items each user will like. We rank each user's unconsumed items by their posterior expected Poisson parameters,

$$\text{score}_{ui} = \mathbb{E}\left[\sum_{k=1}^{\infty}\theta_{uk}\beta_{ik}\right], \tag{18}$$

where $\mathbb{E}[\cdot]$ denotes expectation with respect to the distribution q. During testing, we generate the top M recommendations for each user as those items with the highest predictive score under each method. For each user, we compute the precision-at-M, which measures the fraction of relevant items in the user's top-M recommendations. Likewise, we compute recall-at-M, which is the fraction of items in the test set present in the top M recommendations. We set M to 100 in our experiments. We computed the mean precision and mean recall over $10,000$ randomly chosen users (for MovieLens1M, we compute the mean over all users).

Hyperparameter selection. In order to set the hyperparameters, we first notice that both the finite and infinite models share the same prior on the item weights and, therefore, we use the same hyperparameter values for a fair comparison (specifically, we set $a = b = 0.3$ for both models). Due to the stick breaking construction, the prior on the user weights differs between the finite and the infinite models. For the BNPPF model, we set the user scaling hyperparameter $c = 1$ and $\alpha = 1.1$ (recall from Section 3 that we require $\alpha > 1$). For the finite model, we explored the values in the set $\{0.1, 1, 10\}$ for both the shape and scale in Eq. 2, and choose unit shape and unit scale because these values provided the best performance on the test set in terms of predictive log likelihood (see Supplementary Material for a comparison). We use these values in all our experiments.

Convergence. We terminate the training process when the algorithm converges. We measure convergence by computing the prediction accuracy on a validation set, composed of 1% randomly selected ratings, which are treated as zeros during training and are not considered to measure performance. The algorithm stops either when the change in log likelihood on this validation set is less than 0.0001%, or if the log likelihood decreases for consecutive iterations.

Results. In Figure 2, we show the results for the two MovieLens data sets with 1 million and 10 million rat-

[2]Our software is available at
https://github.com/premgopalan/bnprec.

ings, and the Netflix dataset with 100 million ratings. The top row corresponds to mean held-out log likelihood, while the second and third rows correspond, respectively, to mean precision and recall. We set the truncation level for the BNPPF to $T = 200$ across all data sets. As seen in Figure 2, the BNPPF model has better held-out log likelihood with a fixed truncation level than the corresponding finite models, as we vary K from 10 to 200 (with the exception of the small MovieLens1M dataset). Further, the BNPPF model performs at least as well as the finite models in the mean precision and mean recall metrics.

For the BNPPF model, we computed the effective dimensionality K^* of the latent weights. In order to identify K^*, for each user, we identify the top latent components $k \leq T$, that contribute to at least 95% of the user's expected budget under the approximate posterior, i.e., $\mathbb{E}\left[\sum_{k=1}^{\infty} \theta_{uk} \sum_{i=1}^{M} \beta_{uk}\right]$. We rank the latent components by their contribution to the expected budget. Across all users, it gives us the effective dimensionality of the latent weights. For the MovieLens1M dataset, we found $K^* = 92$, and for MovieLens10M and Netflix data sets, we found that all latent components were used with $T = 200$.

The last row of Figure 2 shows that the BNPPF model runs as fast as the inference algorithm for the finite model with K set to the truncation level of 200. As discussed in Section 3, the dominant computational cost is the same for these algorithms.

The use of degenerate variational distributions over the stick proportions results in a fast algorithm with closed-form solutions for the variational parameters. However, there are two limitations. First, as discussed in Section 3, the choice of degenerate variational distributions constrains α to be greater than 1. We focussed on movie data sets in this work. In future work, we plan on extending our algorithm to other types of data sets, such as users reading scientific articles. Unlike movie data sets, each user is likely to be interested in articles belonging to particular research areas, e.g., machine learning. This suggests increased sparsity in the latent user weights, which can be captured as prior knowledge by setting $\alpha \leq 1$. Second, the use of point estimates for the stick proportions may result in overfitting. Both limitations can be overcome by placing a non-degenerate variational distribution over the stick proportions, which may come at an increased computational cost due to numerical optimization.

5 Conclusion

We develop Bayesian nonparametric Poisson factorization for recommendation systems. Given user/item data, our model captures patterns of preferences and how many components are needed to represent the data. Our efficient variational inference algorithm scales to large data sets with high data sparsity as are frequently encountered in real-world recommendation problems. We present one of the first studies of an infinite model on large-scale user-movie preference data sets. Our model avoids the computationally inefficient method of searching for the number of latent components and gives better predictive performance than its parametric counterpart.

There are several directions for future work. First, we plan to use variational Beta distributions to capture the posterior of the stick proportions. This will remove constraints on settings for hyperparameter α. However, using Beta distributions complicates the variational inference algorithm. Second, some data sets are too large to be handled with batch optimization, even when we only need to process the non-zero entries. We plan to develop stochastic variational inference algorithms (Hoffman et al., 2013) to scale to massive data sets. Finally, in some settings we may want hierarchical structure (Broderick et al., 2013).

Acknowledgements

David M. Blei is supported by NSF IIS-0745520, NSF IIS-1247664, NSF IIS-1009542, ONR N00014-11-1-0651, Alfred P. Sloan foundation, and DARPA FA8750-14-2-0009. Francisco J. R. Ruiz acknowledges Ministerio de Educación of Spain (FPU fellowship, AP2010-5333), University Carlos III in Madrid (Ayuda para la Movilidad, Programa Propio de Investigación), and projects by Ministerio de Ciencia e Innovación of Spain (ALCIT TEC2012-38800-C03-01) and the European Union (Marie Curie ITN, FP7-PEOPLE-2012-ITN-316861). Rajesh Ranganath is supported by an NDSEG fellowship.

References

Ball, B., Karrer, B., & Newman, M. E. J. (2011, September). Efficient and principled method for detecting communities in networks. *Physical Review E*, *84*(3), 036103.

Beal, M. J., & Ghahramani, Z. (2003). The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, *7*.

Broderick, T., Mackey, L., Paisley, J., & Jordan, M. I. (2013). Combinatorial clustering and the beta negative binomial process. *arXiv preprint arXiv:1111.1802*.

Bryant, M., & Sudderth, E. B. (2012). Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *Advances in neu-*

ral information processing systems (NIPS) (pp. 2708–2716).

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* (pp. 122–129). New York, NY, USA: ACM.

Çinlar, E. (1975). *Introduction to stochastic processes* ([Nachdr.] ed.). Englewood Cliffs, NJ: Prentice-Hall.

Cemgil, A. T. (2009, May). Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, *2009*.

Dror, G., Koenigstein, N., Koren, Y., & Weimer, M. (2012). The Yahoo! music dataset and KDD-Cup'11. *JMLR Workshop and Conference Proceedings*, *18*, 3–18.

Dunson, D. B., & Herring, A. H. (2005). Bayesian latent variable models for mixed discrete outcomes. *Biostatistics*, *6*(1), 11–25.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*.

Gantner, Z., Drumond, L., Freudenthaler, C., & Schmidt-Thieme, L. (2012). Personalized ranking for non-uniformly sampled items. *Journal of Machine Learning Research - Proceedings Track*, *18*, 231–247.

Ghahramani, Z., & Beal, M. (2001). Propagation algorithms for variational Bayesian learning. In *Neural information processing systems* (pp. 507–513).

Gopalan, P., Hofman, J. M., & Blei, D. M. (2013). Scalable recommendation with Poisson factorization. *arXiv preprint arXiv:1311.1704*.

Griffiths, T. L., & Ghahramani, Z. (2011). The Indian buffet process: an introduction and review. *Journal of Machine Learning Research*, *12*, 1185-1224.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 230–237). New York, NY, USA: ACM.

Hoffman, M. D., Blei, D. M., & Cook, P. R. (2010). Bayesian nonparametric matrix factorization for recorded music. In J. Frnkranz & T. Joachims (Eds.), *ICML* (pp. 439–446). Omnipress.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *J. Mach. Learn. Res.*, *14*(1), 1303–1347.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 eighth ieee international conference on data mining* (pp. 263–272). Washington, DC, USA: IEEE Computer Society.

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

Knowles, D. A., & Ghahramani, Z. (2011). Nonparametric Bayesian sparse factor models with application to gene expression modelling. *Annals of Applied Statistics*, *5*(2B), 1534–1552.

Koren, Y., Bell, R., & Volinsky, C. (2009, August). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Kurihara, K., Welling, M., & Vlassis, N. (2006). Accelerated variational Dirichlet process mixtures. In *Advances in neural information processing systems (NIPS).*

Liang, P., Petrov, S., Jordan, M. I., & Klein, D. (2007). The infinite PCFG using hierarchical Dirichlet processes. In *Empirical methods in natural language processing* (pp. 688–697).

Miller, K. (2011). *Bayesian nonparametric latent feature models.* Unpublished doctoral dissertation, EECS Department, University of California, Berkeley, CA. (UCB/EECS-2011-78)

Porteous, I., Bart, E., & Welling, M. (2008). Multi-HDP: A nonparametric Bayesian model for tensor factorization. In D. Fox & C. P. Gomes (Eds.), *Aaai* (pp. 1487–1490). AAAI Press.

Salakhutdinov, R., & Mnih, A. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems* (Vol. 20).

Teh, Y. W., & Jordan, M. I. (2010). Hierarchical Bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, & S. Walker (Eds.), *Bayesian nonparametrics: Principles and practice.* Cambridge University Press.

Thibaux, R. (2008). *Nonparametric Bayesian models for machine learning.* Unpublished doctoral dissertation, EECS Department, University of California, Berkeley, CA. (UCB/EECS-2008-130)

Titsias, M. (2007). The infinite gamma-Poisson feature model. *Advances in Neural Information Processing Systems (NIPS)*, *19*.

Zhou, M., & Carin, L. (2013). Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *99*, 1.

Zhou, M., Hannah, L., Dunson, D. B., & Carin, L. (2012). Beta-negative binomial process and Poisson factor analysis. *Journal of Machine Learning Research - Proceedings Track*, *22*, 1462–1471.