Recommendation from Raw Data with Adaptive Compound Poisson Factorization

Olivier Gouvert, Thomas Oberlin, Cédric Févotte
IRIT, Université de Toulouse, CNRS, France
firstname.lastname@irit.fr

Abstract

Count data are often used in recommender systems: they are widespread (song play counts, product purchases, clicks on web pages) and can reveal user preference without any explicit rating from the user. Such data are known to be sparse, over-dispersed and bursty, which makes their direct use in recommender systems challenging, often leading to pre-processing steps such as binarization. The aim of this paper is to build recommender systems from these raw data, by means of the recently proposed compound Poisson Factorization (cPF). The paper contributions are three-fold: we present a unified framework for discrete data (dcPF), leading to an adaptive and scalable algorithm; we show that our framework achieves a trade-off between Poisson Factorization (PF) applied to raw and binarized data; we study four specific instances that are relevant to recommendation and exhibit new links with combinatorics. Experiments with three different datasets show that dcPF is able to effectively adjust to over-dispersion, leading to better recommendation scores when compared with PF on either raw or binarized data.

1 INTRODUCTION

Collaborative filtering (CF) techniques have been achieving state-of-the-art performances in recommendation tasks since the Netflix prize (Bennett and Lanning, 2007). CF is based on feedbacks of users interacting with items. These data can either be explicit (ratings, thumbs up/down) or implicit (number of times a user listened to a song, number of clicks on web pages). In particular, historical data are easy to collect and often in the form of count data. They can be stored in a sparse matrix $\mathbf{Y}$ of size $U \times I$, where each entry of the matrix y_{ui} is the number of times the user $u \in \{1, \ldots, U\}$ interacts with the item $i \in \{1, \ldots, I\}$. For the rest of the paper, we will consider the example of users listening to songs without loss of generality.

Matrix factorization (MF) allows to make recommendations using these feedback data (Koren et al., 2009). The aim of MF is to infer a low-rank approximation of the observations: $\mathbf{Y} \approx \mathbf{W}\mathbf{H}^T$, where $\mathbf{W}$ of size $U \times K$ represents the preferences of users, and $\mathbf{H}$ of size $I \times K$ represents the attributes of items, with $K \ll \min(U, I)$. Therefore, each user or item is represented in the same latent space by a vector of K latent components. The strength of an interaction between a user and an item is measured by the dot product between their representative latent vectors. Among the methods based on MF (Lee and Seung, 1999, 2001; Hu et al., 2008; Ma et al., 2011; Févotte and Idier, 2011; Liang et al., 2016), Poisson factorization (PF) (Canny, 2004; Cemgil, 2009; Gopalan et al., 2015) has become very popular in CF when using implicit feedbacks. Indeed, PF posits that the data are generated from a Poisson distribution, making it well-adapted for count data. PF has reached state-of-the-art results while having favorable properties. (i) PF down-weighs the effect of the zeros present in the data, by implicitly assuming that the users have a limited budget to distribute among the items (Gopalan et al., 2015). (ii) Algorithms for PF scale with the number of non-zero values in the data, leading to fast inference (Cemgil, 2009). Many variants of PF have been proposed these last years. Hierarchical structures on the latent variables have been explored (Ranganath et al., 2015; Zhou et al., 2015; Liang et al., 2018). Other works have proposed to use additional information in the model to perform hybrid CF approaches (Gopalan et al., 2014; Lu et al., 2018; Salah and Lauw, 2018).

However, in many cases, count data are over-dispersed and bursty (Kleinberg, 2003; Schein et al., 2016). The Poisson distribution fails to fully describe such data. Its

modeling capacities are indeed limited since its mean and variance are equal. To avoid this problem, it is of common use to work with binarized data (Gopalan et al., 2015; Liang et al., 2016). This pre-processing step is effective in practice but removes the information contained in the non-zero values. Recent works have focused on directly using the raw data in order to achieve better representation and recommendation results. In (Hu et al., 2008; Pan et al., 2008), the raw data are introduced as weights (confidence), which regularize the MF approximation. Other works try to find generative processes which are able to deal with over-dispersed data. In (Zhou, 2018), the author makes use of the negative binomial (NB) distribution, which is a well-known extension of the Poisson distribution (Lawless, 1987). He exploits the compound Poisson (cP) representation of the NB distribution to preserve the scalability property of the proposed algorithm. cP structure has further been used in (Simsekli et al., 2013; Basbug and Engelhardt, 2016) to model continuous or discrete sparse data, showing an improved description of the non-zero values.

In this paper, we present novel contributions to discrete compound Poisson factorization (dcPF). dcPF refers to compound Poisson factorization (cPF), as introduced by (Basbug and Engelhardt, 2016), for discrete data. dcPF posits that the listening counts can be grouped in listening sessions which are somewhat more informative for recommendation. It uses the concept of self-excitation (Du et al., 2015; Hosseini et al., 2018; Khodadadi et al., 2018; Zhou, 2018), which describes the idea that a user can listen to a song not merely because of his/her attachment to it, but because of a previous interaction. The contributions of the paper are the following:

• We develop a unified framework for dcPF and study four specific distributions to model self-excitation, called element distributions. We exhibit new links between the choice of this distribution and combinatorics.

• We provide simple conditions to preserve scalability and to obtain closed-form updates for the inference of the posterior.

• We show that dcPF is a natural generalization of PF by proving that PF applied to raw data and PF applied to binarized data are two limit cases of dcPF.

• We discuss the choice of the element distribution and in particular consider a new one in the context of compound Poisson models, the shifted negative binomial distribution. We present new methodology for hyper-parameter estimation and report experiments with three datasets.

The paper is organized as follows. In Section 2, we provide preliminary material about PF and exponential dispersion models (EDM). In Section 3, we present Bayesian

dcPF and give an intuitive interpretation of the model and its properties. Related works are discussed in Section 4 and a scalable variational algorithm is developed in Section 5. In Section 6, we apply the proposed algorithm to recommendation tasks with three datasets. Section 7 concludes the paper and discusses possible perspectives.

2 PRELIMINARIES

Poisson factorization. PF is based on non-negative matrix factorization (NMF) (Lee and Seung, 1999, 2001). Each observation is assumed to be drawn from a Poisson distribution:

$$y_{ui} \sim \text{Poisson}([\mathbf{WH}^T]_{ui}), \qquad (1)$$

with $y_{ui} \in \mathbb{N} = \{0, 1, \ldots, +\infty\}$. The preferences $\mathbf{W}$ and the attributes $\mathbf{H}$ are supposed to be non-negative matrices. Non-negativity induces a constructive *part-based* representation of the observations that is central to so-called *topic models* (Lee and Seung, 1999; Blei et al., 2003). Bayesian extensions of PF typically impose that each entry of the matrices $\mathbf{W}$ and $\mathbf{H}$ has a gamma prior. The gamma prior[1] imposes non-negativity and is known to induce sparsity when the shape parameter is lower than one. This is a desirable property in the sense that it implies that users and items are represented by only a few patterns. Moreover, it is conjugate with the Poisson distribution, which proves convenient for variational inference.

PF has been very popular in the last decade because of its scalability with sparse data. Sparsity is very common in recommender systems, since subsets of users usually interact with only subsets of items from a large catalog. Using the superposition property of the Poisson distribution, we can augment the model presented in Equation (1) as follows (Cemgil, 2009; Gopalan et al., 2015):

$$y_{ui} = \sum_k c_{uik}; \ c_{uik} \sim \text{Poisson}(w_{uk}h_{ik}). \qquad (2)$$

The conditional distribution of this new latent variable follows a multinomial distribution: $\mathbf{c}_{ui}|y_{ui} \sim \text{Mult}(y_{ui}, \boldsymbol{\phi}_{ui})$, where $\boldsymbol{\phi}_{ui}$ is a vector of size K with entries $\phi_{uik} = \frac{w_{uk}h_{ik}}{[\mathbf{WH}^T]_{ui}}$. The latent variable $\mathbf{c}_{ui}$ is central to state-of-the-art PF algorithms. $y_{ui} = 0$ implies that $\mathbf{c}_{ui} = \mathbf{0}_K$, where $\mathbf{0}_K$ is a vector of size K full of zeros. As such, the latent variable $\mathbf{c}_{ui}$ only needs to be estimated for the non-zero values of $\mathbf{Y}$, which ensures scalability provided the data is sparse.

One limitation of the Poisson distribution is that its variance is equal to its mean: $var(y_{ui}) = \langle y_{ui} \rangle$. This makes

---

[1] We use the following convention for the gamma distribution: $\mathcal{G}(x; \alpha, \beta) = x^{\alpha-1} e^{-\beta x} \beta^\alpha \Gamma(\alpha)^{-1}$ where α is the shape parameter and β is the rate parameter.

it ill-suited for over-dispersed data. Moreover, when working with raw data, it appears that PF does not correctly weigh the observations and is too sensitive to large values. The Poisson distribution, parametrized by only one parameter, thus appears too restrictive to model both sparse and heavy-tailed data. To circumvent these issues, data binarization is often used as pre-processing (Gopalan et al., 2015; Liang et al., 2016), with the loss of information it induces. The goal of dcPF studied in this paper is to preserve the data while accounting for sparsity and over-dispersion in the model. When necessary, we denote by $\mathbf{Y}^b$ the corresponding binary version of the observations, where $y_{ui}^b = \mathbb{1}[y_{ui} > 0]$.

Exponential dispersion model. A central element of cP models is the distribution used to model the self-excitation. In this paper, we will assume that it belongs to the well-studied EDM family (Jørgensen, 1986, 1987). It is a convenient choice when dealing with cP models (Yılmaz and Cemgil, 2012; Simsekli et al., 2013; Basbug and Engelhardt, 2016), as explained next. Most discrete random variables can be written in the form of a discrete EDM, denoted by $x \sim ED(\theta, \kappa)$, and defined by

$$p(x; \theta, \kappa) = \exp(x\theta - \kappa\psi(\theta))h(x, \kappa), \ x \in S_\kappa, \quad (3)$$

where $\theta \in \Theta \subset \mathbb{R}$ is called the natural parameter, $\kappa > 0$ is called the dispersion parameter, $\psi(\theta)$ is the log-partition function, $h(x, \kappa)$ is the base measure and S_κ is the support of the distribution, which depends of κ. The mean of x is given by $\langle x \rangle = \kappa\psi'(\theta)$ and its variance by $var(x) = \kappa\psi''(\theta)$. One of the most interesting properties of EDM is the property of additivity. If $x_l \sim ED(\theta, \kappa)$ and $y = \sum_{l=1}^n x_l$ with $n \in \mathbb{N}$, then $y \sim ED(\theta, n\kappa)$. By convention, we assume that $ED(\theta, 0)$ is a Dirac distribution in 0.

3 BAYESIAN DISCRETE COMPOUND POISSON FACTORIZATION

3.1 MODEL DESCRIPTION

We consider the framework proposed by (Basbug and Engelhardt, 2016). The generative model of the observations $\mathbf{Y}$ is given by

$$w_{uk} \sim \mathcal{G}(\alpha^W, \beta_u^W), \ h_{ik} \sim \mathcal{G}(\alpha^H, \beta_i^H), \quad (4)$$

$$n_{ui} \sim \text{Poisson}\left([\mathbf{WH}^T]_{ui}\right), \quad (5)$$

$$x_{l,ui} \sim ED(\theta, \kappa), \ \forall l \in \{1, \ldots, n_{ui}\}, \quad (6)$$

$$y_{ui} = \sum_{l=1}^{n_{ui}} x_{l,ui}. \quad (7)$$

We here specifically assume that $x_{l,ui}$ is a discrete random variable with support equal to $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. The

rate parameters of the gamma priors are treated as deterministic parameters estimated by maximum likelihood (ML). $\boldsymbol{\beta}^W \in \mathbb{R}_+^U$ expresses the activity level of the users and $\boldsymbol{\beta}^H \in \mathbb{R}_+^I$ expresses the popularity of the items. A Bayesian treatment of these parameters is also possible and is considered in (Gopalan et al., 2015). Using the additivity property of EDM, we can easily marginalize the latent variables $x_{l,ui}$, leading to: $y_{ui} \sim ED(\theta, n_{ui}\kappa)$. Compared to PF, this additional stage in the generative process allows for a flexible description of the observations. There are two additional parameters $\{\theta, \kappa\}$ which control the variance and tail of the distribution.

Interpretation. In this paragraph, we will suppose that a user/item pair is fixed. For conciseness, we will omit the corresponding indices $_{ui}$. dcPF introduces new latent variables n and $\{x_l\}_l$ for $l \in \{1, \ldots, n\}$. The latent variable n represents the number of listening sessions the user has had for the song. During each session, indexed by l, the user listened to the song a number of times x_l which is greater or equal to one. The latent variable x_l models the self-excitation induced by a listening interaction. This concept has been used in (Zhou, 2018; Hosseini et al., 2018). Thus, a user can listen to a song, not merely because he/she likes it, but because of a previous listening/excitation. For example, a user can have a summer crush for a song and may listen to it on repeat. The first listening reflects the interest of the user for this song, whereas the following listenings are the consequence of the first one and reflect a short-term behavior. Therefore these listening counts can be grouped in a few listening sessions that will be more able to represent long-term preferences. Finally, the observed variable y is just the aggregation of all the listening counts from all the sessions. The variable n can be viewed as a way to partition the observation y in a smaller number of sessions. This number n better reflects the preferences of the user, since it is deprived of the notion of self-excitation which artificially inflates the number of listening counts. In the following, we will denote by $\mathbf{N} \in \mathbb{N}^{U \times I}$ the *exposition matrix* with entries $[\mathbf{N}]_{ui} = n_{ui}$.

Joint log-likelihood. The joint log-likelihood of the observations $\mathbf{Y}$ and of the latent variables $\mathbf{N}$, $\mathbf{W}$ and $\mathbf{H}$ can be written as follows:

$$\log p(\mathbf{Y}, \mathbf{N}, \mathbf{W}, \mathbf{H}) = \underbrace{\log p(\mathbf{Y}|\mathbf{N}; \theta, \kappa)}_{\text{Mapping}} \quad (8)$$
$$+ \underbrace{\log p(\mathbf{N}|[\mathbf{WH}^T])}_{\text{PF structure}} + \underbrace{\log p(\mathbf{W}, \mathbf{H})}_{\text{Regularization}}.$$

We can decompose this log-likelihood into three terms: a probabilistic mapping term corresponding to the compound structure of the observations, a term corresponding

to the PF structure on the latent variable $\mathbf{N}$ and a regularization term induced by the gamma priors. Contrary to PF, the factorization is placed on the latent variable $\mathbf{N}$ instead of the data itself, allowing for more flexibility. Going back to our interpretation, this latent variable is more likely to inform on user preference than $\mathbf{Y}$. The mapping term can be viewed as a distortion of the true observations, making them "more factorizable" than the raw observations. Therefore, this additional term allows to avoid strong pre-processing stages (such as binarization), letting the data choose their "own distortion".

Scalability and tractability. By imposing that users will listen to a song at least one time during each session, i.e., $x_{l,ui} \in \mathbb{N}^*$, two important properties can be deduced.

First, we have the following equivalence: $y_{ui} = 0 \Leftrightarrow n_{ui} = 0$. In other words, the observed listening count is equal to zero if and only if the number of listening sessions is equal to zero. Therefore, the latent variable $\mathbf{N}$ is partially known and has the same zeros as $\mathbf{Y}$. Thanks to this, we preserve the scalability property of PF (cf Section 2). Moreover, we have that:

$$\mathbb{P}(y_{ui} = 0) = \mathbb{P}(n_{ui} = 0) = e^{-[\mathbf{W}\mathbf{H}^T]_{ui}}. \qquad (9)$$

The latent variables $\mathbf{W}$ and $\mathbf{H}$ control the sparsity of the matrix $\mathbf{Y}$, while the element distribution and its parameters $\{\theta, \kappa\}$ only focus on the representation of non-zero values.

The second interesting property is that $n_{ui} \leq y_{ui}$. Therefore, given an observation y_{ui}, we know that n_{ui} can only take a finite number of values, bounded by y_{ui} (in particular, $y_{ui} = 1 \Rightarrow n_{ui} = 1$). This provides efficient means of calculation for the latent variable $\mathbf{N}$ during inference.

3.2 EXAMPLES OF ELEMENT DISTRIBUTIONS

Distributions based on Stirling numbers. In this paragraph we focus on three particular distributions: the logarithmic distribution (Quenouille, 1949), denoted by $x_l \sim Log(p)$; the zero-truncated Poisson (ZTP) distribution, denoted by $x_l \sim ZTP(p)$; the (shifted) geometric distribution, denoted by $x_l \sim Geo(1 - p)$.[2] Examples of probabilistic mass functions (p.m.f.) of the four considered element distributions are displayed on Figure 1.

These three distributions can be written in the form of a discrete EDM with dispersion parameter $\kappa = 1$ and support $\mathbb{N}^*$. Their base measure is given by $h(x_l, \kappa) = \frac{x_l!}{\kappa!} St_j(x_l, \kappa)$, where $St_j(x_l, \kappa)$ is the unsigned Stirling number of one of the three kinds ($j \in \{1, 2, 3\}$), see
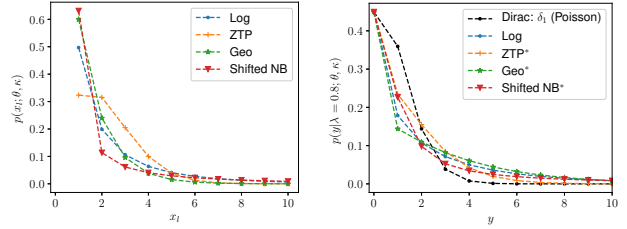

Figure 1: On the left, the p.m.f. of the four element distributions presented in Section 3.2. On the right, the p.m.f. of the marginalized distribution of the observations, using the same four element distributions. The values of the natural and dispersion parameters are those presented in Table 3. The p.m.f. marked with a * are not available in closed form and are represented through a histogram of simulated values.

Table 1. See (Johnson et al., 2005) for more details. It is of particular interest when analyzing the distribution of $y|n$. This conditional distribution is also a discrete EDM with: $h(y, \kappa n) = \frac{y!}{n!} St_j(y, n)$.

The Stirling numbers of the three kinds are three different ways to partition y elements into n groups (Riordan, 2012) (graphical illustrations are given in the supplementary material):

• The Stirling number of the first kind corresponds to the number of ways of partitioning y elements into n disjoints cycles. It can be obtained thanks to a recurrence formula: $St_1(y + 1, n) = y\,St_1(y, n) + St_1(y, n - 1)$.

• The Stirling number of the second kind corresponds to the number of ways of partitioning y elements into n non-empty subsets. It can be calculated in closed form: $St_2(y, n) = \frac{1}{n!} \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} j^y$. When y is too large, its exact computation can suffer from numerical issues, though reasonable and stable approximations are available (Bleick and Wang, 1974).

• The Stirling number of the third kind (also known as Lah number) corresponds to the number of ways of partitioning y elements into n non-empty ordered subsets. It is given by: $St_3(y, n) = \binom{y-1}{n-1} \frac{y!}{n!}$. Its definition is particularly well adapted if we assume that the grouping results from temporal phenomena.

Shifted negative binomial. The dispersion parameter for the three distributions presented in the previous examples is fixed and equal to one. We now introduce a new distribution, referred to as *shifted NB* distribution which is parametrized by two parameters: $x_l - 1 \sim NB(a, p)$, whose shape parameter a controls the long tail of the distribution, and $p \in (0, 1)$ is the probability parameter. The shifted NB is a shifted

Table 1: Examples of four discrete element distributions. Notation: $\mathbb{R}_-^* = (-\infty; 0)$.

Distribution	θ	Θ	θ^{raw}	θ^{bin}	κ	$\psi(\theta)$	$h(x, \kappa)$
$x_l \sim \text{Log}(p)$	$\log(p)$	$\mathbb{R}_-^*$	$-\infty$	0	1	$\log(-\log(1 - e^\theta))$	$\frac{x!}{\kappa!} St_1(x, \kappa)$
$x_l \sim \text{ZTP}(p)$	$\log(p)$	$\mathbb{R}$	$-\infty$	$+\infty$	1	$\log(e^{e^\theta} - 1)$	$\frac{x!}{\kappa!} St_2(x, \kappa)$
$x_l \sim \text{Geo}(1-p)$	$\log(p)$	$\mathbb{R}_-^*$	$-\infty$	0	1	$\log(\frac{e^\theta}{1 - e^\theta})$	$\frac{x!}{\kappa!} St_3(x, \kappa)$
$x_l - 1 \sim \text{NB}(a, p)$	$\log(p)$	$\mathbb{R}_-^*$	$-\infty$	0	$(1, a)^T$	$(\theta, -\log(1 - e^\theta))^T$	$\frac{\Gamma(x - \kappa_1 + \kappa_2)}{\Gamma(x - \kappa_1 + 1)\Gamma(\kappa_2)}$

EDM, which does not exactly fall into the EDM family. However, the conditional distribution $y|n$ can still be written as: $p(y|n; \theta, \kappa) = \exp(y\theta - n\kappa^T \psi(\theta)) h(y, n\kappa)$, where $y \in \{n, \dots, +\infty\}$, $\kappa = (\kappa_1, \kappa_2)^T = (1, a)^T$ and $\psi(\theta) = (\theta, -\log(1 - e^\theta))^T$. Note that κ and $\psi(\theta)$ are now vectors of dimension 2. Parameter κ_1 controls the shifting operation, and is fixed to one to ensure that the support of x_l is $\mathbb{N}^*$. Shifted NB encompasses two particular cases: the classical NB distribution ($\kappa_1 = 0$), and the geometric distribution ($\kappa_1 = \kappa_2 = 1$).

For each distribution, the resulting marginalized distribution of y is displayed on Figure 1 and is compared to the Poisson distribution (which is a cP distribution with a Dirac as element distribution). We can see on this figure that all marginalized distributions have the same mass in 0 (see Section 3.1) but are different otherwise.

3.3 A TRADE-OFF BETWEEN RAW AND BINARIZED DATA

In this section, we show that dcPF generalizes PF in the sense that it includes PF applied to raw and binarized data as limit cases. For a given dispersion parameter κ, the natural parameter θ controls the level of information contained in the observations $\mathbf{Y}$:

• When θ tends to a limit θ^{raw}, dcPF becomes equivalent to PF (with original raw data).

• When θ tends to a limit θ^{bin}, the posterior inference of dcPF becomes equivalent to the posterior inference of PF applied to the binarized data. In other words, performing dcPF (with original raw data) becomes equivalent to performing PF on binarized data. Note that the marginal distribution of the observations and the distribution of $y|n$ are both improper distributions, but the posterior distribution is still well-defined (Robert, 2007).

• Between θ^{raw} and θ^{bin}, θ controls the degree of *implicit distortion* of the observations.

Our results are formalized in the two following propositions. The proofs are left to the supplementary material.

Proposition 1. *If there exists θ^{raw} such that $\lim_{\theta \to \theta^{raw}} \kappa^T \psi(\theta) = -\infty$, then the posterior of*

dcPF tends to the posterior of PF as θ goes to θ^{raw}.

Proposition 2. *If there exists θ^{bin} such that $\lim_{\theta \to \theta^{bin}} \kappa^T \psi(\theta) = +\infty$, then the posterior of dcPF tends to the posterior of PF applied to binarized data as θ goes to θ^{bin}, i.e.: $\lim_{\theta \to \theta^{bin}} p(\mathbf{W}, \mathbf{H}|\mathbf{Y}) = p(\mathbf{W}, \mathbf{H}|\mathbf{N} = \mathbf{Y}^b)$.*

The four distributions described in Section 3.2 respect the assumptions of both propositions. The limit cases of the natural parameter θ are given in Table 1.

It is of particular interest to learn the natural parameter θ since its choice characterizes the data. If θ is close to θ^{raw}, the observations do not need to be distorted and PF on raw data is effective. If θ is close to θ^{bin}, the non-zero observations of $\mathbf{Y}$ are non-informative and binarization is welcome. In between these extremes, dcPF takes full power and acts as an implicit distortion. Thus, the value of θ gives an indication on the gain brought by dcPF as compared to PF.

4 RELATED WORKS

Negative binomial factorization. An extension of the Poisson distribution known to model over-dispersion is the NB distribution. The NB distribution depends on two parameters: a shape parameter and a probability parameter p. In (Zhou, 2018), the author introduces NB matrix factorization, in which he posits that the shape parameter is low-rank, i.e.: $y_{ui} \sim NB([\mathbf{WH}^T]_{ui}, p)$. To preserve scalability of the proposed Gibbs algorithm, the author uses the cP representation of the NB (Quenouille, 1949; Fisher et al., 1943): $n_{ui} \sim \text{Poisson}(-[\mathbf{WH}^T]_{ui} \log(1 - p))$ and $y_{ui} \sim \text{SumLog}(n_{ui}, p)$ where $\text{SumLog}(n, p)$ is the sum of n identical and independent logarithmic distributions. In this case, the conditional distribution of the n_{ui} is also known: $n_{ui}|y_{ui} \sim CRT(y_{ui}, [\mathbf{WH}^T]_{ui})$, where CRT is the number of opened tables in a Chinese restaurant process (CRP). An important difference with our framework is that the parameter p there controls both the sparsity of $\mathbf{Y}$ and the distribution of the non-zero values. This introduces a coupling between the factorization $\mathbf{WH}^T$ and the parameter p which leads to a more difficult interpretation in the context of recommendation.

Compound Poisson models. In (Basbug and Engelhardt, 2016), the authors introduce cPF which is well-adapted for continuous or discrete sparse data. For discrete data, the authors present four different distributions but only one (the ZTP distribution) with support $\mathbb{N}^*$. Note that, if $\mathbb{P}(x_{l,ui} = 0) > 0$ then the latent variable $\mathbf{N}$ is completely unknown and the scalability property does not hold anymore (unless the hypothesis $y_{ui} = 0 \Rightarrow n_{ui} = 0$ is arbitrarily imposed during the inference). In terms of inference, (Basbug and Engelhardt, 2016) describe a stochastic variational inference algorithm that is shown to perform well in terms of log-likelihood computed from held-out data. We will instead evaluate the performance of dcPF with recommendation metrics.

In (Yılmaz and Cemgil, 2012; Simsekli et al., 2013), a cP structure with a gamma element distribution is used to represent the Tweedie distribution. The Tweedie distribution is the distribution induced by the β-divergence with $\beta \in (01)$ (Févotte and Idier, 2011). One importance difference with our setting, besides the fact that the Tweedie distribution is continuous, is that the authors impose that the model is mean-parametrized, i.e., $\langle y_{ui} \rangle = [\mathbf{WH}^T]_{ui}$. This is not the case with cPF since by construction: $\langle y_{ui} \rangle \geq \langle n_{ui} \rangle = [\mathbf{WH}^T]_{ui}$.

Weighted MF. In (Hu et al., 2008), the authors develop a framework for implicit feedbacks. Implicit data are inherently noisy and may not reflect a direct preference of a user for an item, but rather a confidence in the interaction. In this context, implicit feedbacks can be transformed and incorporated as weights in the cost function which is defined as:

$$C(\mathbf{W}, \mathbf{H}) = \sum_{ui} \omega_{ui} \|y_{ui}^b - [\mathbf{WH}^T]_{ui}\|_2^2 + \mu R(\mathbf{W}, \mathbf{H}),$$

where $\omega_{ui} = f(y_{ui})$ is the confidence that can be brought in the binary observation y_{ui}^b, f is a fixed mapping function, $R(\mathbf{W}, \mathbf{H})$ is a regularization term and μ is an hyper-parameter. Here, the mapping function f between the raw data and the confidence is deterministic. Note that some other works focused on introducing probabilistic weights in the data fitting term (Liang et al., 2016; Wang et al., 2018) but these weights are learned regardless of the raw data. As discussed previously, dcPF encompasses the raw observations via an additional probabilistic mapping term. This term can also be viewed as a probabilistic confidence term, combining the two latter approaches. Indeed, large listening counts y_{ui} will often lead to a large number of sessions n_{ui}, exhibiting a strong confidence in this observation. Nevertheless, this mapping is not deterministic and as such more flexible and robust.

5 VARIATIONAL BAYES EXPECTATION-MAXIMIZATION

In this section, we develop a variational Bayes expectation-maximization (VBEM) algorithm. We denote by $\mathbf{Z} = \{\mathbf{N}, \mathbf{C}, \mathbf{W}, \mathbf{H}\}$ the set of latent variables and by $\Phi = \Phi_1 \cup \Phi_2$ the set of parameters, with $\Phi_1 = \{\theta, \kappa\}$ and $\Phi_2 = \{\alpha^W, \boldsymbol{\beta}^W, \alpha^H, \boldsymbol{\beta}^H\}$. The aim of this algorithm is to estimate both the posterior $p(\mathbf{Z}|\mathbf{Y}; \Phi)$ and the parameters Φ.

5.1 VARIATIONAL INFERENCE

Bayesian inference revolves around the characterization of the posterior distribution $p(\mathbf{Z}|\mathbf{Y}; \Phi)$. Unfortunately, this posterior is intractable in our case. Variational inference (VI) (Jordan et al., 1999; Blei et al., 2017) consists in approximating this intractable posterior by a simpler distribution q parametrized by its own parameters $\tilde{\Phi}$, called variational parameters. Thus, the aim of VI is to minimize the Kullback-Leibler divergence between the true and approximate distributions with respect to (w.r.t.) the variational parameters. In practice, it is simpler to maximize the so-called expected lower bound (ELBO), which is an equivalent problem. A common choice is to assume q to be factorizable (mean-field approximation):

$$q(\mathbf{Z}) = \prod_{ui} q(n_{ui}, \mathbf{c}_{ui}) \prod_{uk} q(w_{uk}) \prod_{ik} q(h_{ik}). \quad (10)$$

Though not explicitly shown for conciseness, the variational distribution of each parameter is governed by its own set of parameters (over which optimization takes place). Note that we choose the latent variables n_{ui} and $\mathbf{c}_{ui}$ to remain coupled. We can further decompose the variational distribution of these variables as: $q(n_{ui}, \mathbf{c}_{ui}) = q(\mathbf{c}_{ui}|n_{ui})q(n_{ui})$.

The ELBO can be calculated as follow:

$$\mathcal{L}(q, \Phi) = \langle \log p(\mathbf{Y}|\mathbf{N}; \Phi_1) \rangle_q + \langle \log p(\mathbf{N}, \mathbf{C}|\mathbf{W}, \mathbf{H}) \rangle_q$$
$$+ \langle \log p(\mathbf{W}, \mathbf{H}; \Phi_2) \rangle_q + \mathcal{H}(q), \quad (11)$$

where $\langle x \rangle_q$ is the expectation of the variable x w.r.t. the variational distribution q and $\mathcal{H}(q)$ is the entropy of the distribution q.

Coordinate ascent VI. We use a coordinate ascent for VI (CAVI) algorithm to maximize the ELBO. The CAVI algorithm consists of sequentially optimizing each of the variational parameters while keeping the others fixed. It can be shown that mean-field variational inference naturally leads to the following choice of variational distributions (Bishop, 2006), parametrized

by $\tilde{\Phi} = \{\Lambda, \tilde{\alpha}^W, \tilde{\beta}^W, \tilde{\alpha}^H, \tilde{\beta}^H\}$:

$$q(w_{uk}) = \mathcal{G}(\tilde{\alpha}_{uk}^W, \tilde{\beta}_{uk}^W), \ q(h_{ik}) = \mathcal{G}(\tilde{\alpha}_{ik}^H, \tilde{\beta}_{ik}^H), \quad (12)$$

$$q(\mathbf{c}_{ui}|n_{ui}) = \mathrm{Mult}\left(n_{ui}, \left\{\frac{\Lambda_{uik}}{\Lambda_{ui}}\right\}_k\right),$$

$$q(n_{ui} = n) = \frac{1}{Z_{ui}}(r_{ui})^n \frac{h(y_{ui}, n\kappa)}{n!}, \forall n \in \{1, \ldots, y_{ui}\},$$

where $\Lambda_{ui} = \sum_k \Lambda_{uik}$, $r_{ui} = \Lambda_{ui}e^{-\kappa\psi(\theta)}$ and $Z_{ui} = \sum_{n=1}^{y_{ui}}(r_{ui})^n \frac{h(y_{ui}, n\kappa)}{n!}$ is a normalization constant.

Update rules. CAVI leads to the following set of iterative update rules:

$$\Lambda_{uik} \leftarrow \exp\left(\langle\log w_{uk}\rangle_q + \langle\log h_{ik}\rangle_q\right); \quad (13)$$

$$\tilde{\alpha}_{uk}^W \leftarrow \alpha^W + \sum_i \langle n_{ui}\rangle_q \frac{\Lambda_{uik}}{\Lambda_{ui}}; \tilde{\beta}_{uk}^W \leftarrow \beta_u^W + \sum_i \langle h_{ik}\rangle_q$$

$$\tilde{\alpha}_{ik}^H \leftarrow \alpha^H + \sum_u \langle n_{ui}\rangle_q \frac{\Lambda_{uik}}{\Lambda_{ui}}; \tilde{\beta}_{ik}^H \leftarrow \beta_i^H + \sum_u \langle w_{uk}\rangle_q.$$

When $x \sim \mathcal{G}(\alpha, \beta)$, $\langle x\rangle = \frac{\alpha}{\beta}$ and $\langle\log x\rangle = \Psi(\alpha) - \log\beta$, where Ψ is the digamma function. The statistic $\langle n_{ui}\rangle_q$ is available in closed form[3] thanks to the properties of Section 3.1. If $y_{ui} = 0$ then $\langle n_{ui}\rangle_q = 0$, otherwise $\langle n_{ui}\rangle_q = \sum_{n=1}^{y_{ui}} n\, q(n_{ui} = n)$.

As expected, we recover, as limit cases, the algorithms for PF (Gopalan et al., 2015) applied to raw data if $\langle n_{ui}\rangle_q = y_{ui}$ and to binarized data if $\langle n_{ui}\rangle_q = \mathbb{1}[y_{ui} > 0]$. The algorithm is stopped when the relative increment of the ELBO gets lower than a value τ.

5.2 PARAMETERS ESTIMATION

Activity and popularity parameters. Optimizing the ELBO w.r.t. the parameters Φ_2 is equivalent to solving the sub-problem: $\mathrm{argmax}_{\Phi_2}\langle\log p(\mathbf{W}, \mathbf{H}; \Phi_2)\rangle_q$. In this article, we suppose that the shape parameters $\{\alpha^W, \alpha^H\}$ are known and we want to optimize only the rate parameters $\{\beta^W, \beta^H\}$. The interested reader is referred to (Cemgil, 2009) and (Zhou and Carin, 2015) for the details of ML and Bayesian estimation of the shape parameter of a gamma distribution. ML leads to the following updates for both activity and popularity parameters:

$$\beta_u^W \leftarrow \frac{\sum_k \langle w_{uk}\rangle_q}{K\alpha^W}; \ \beta_i^H \leftarrow \frac{\sum_k \langle h_{ik}\rangle_q}{K\alpha^H}. \quad (14)$$

Natural parameter. Optimizing the ELBO w.r.t. the natural parameter θ is equivalent to maximizing:

---

[3]When choosing the logarithmic distribution as the element distribution we have: $\langle n_{ui}\rangle_q = r_{ui}\big(\Psi(y_{ui} + r_{ui}) - \Psi(r_{ui})\big)$ (Zhou, 2018).

$\langle\log p(\mathbf{Y}|\mathbf{N}; \Phi_1)\rangle_q = \sum_{ui}(y_{ui}\theta - \langle n_{ui}\rangle_q \kappa\psi(\theta)) + cst$, where cst is a constant w.r.t. the natural parameter θ. It leads to the following equation:

$$\sum_{ui} y_{ui} - \sum_{ui} \langle n_{ui}\rangle_q \kappa\psi'(\theta) = 0. \quad (15)$$

In the case of the shifted NB distribution, $\psi(\theta) \in \mathbb{R}^2$ and ψ' corresponds to its gradient. The solution of this equation in known in closed form for geometric and shifted NB distributions. We implement a Newton-Raphson algorithm to solve it for logarithmic and ZTP distributions.

Dispersion parameter for shifted NB. When choosing the shifted NB as the element distribution, we have:

$$y_{ui} - n_{ui} \sim NB(\kappa_2 n_{ui}, e^\theta). \quad (16)$$

Optimizing the ELBO w.r.t. the parameter κ_2 which controls the long-tail of the NB distribution is not straightforward. The main issue is that it involves a term of the form $\langle h(y_{ui}, n_{ui}\kappa)\rangle_q$ that is computationally expensive to optimize. Therefore, we augment the model like in (Zhou, 2018), with a latent variable: $m_{ui}|y_{ui}, n_{ui} \sim CRT(y_{ui} - n_{ui}, n_{ui}\kappa_2)$ (if $y_{ui} = 0$ then $m_{ui} = 0$). In this augmented model, the optimization of κ_2 is equivalent to finding the ML estimator of $m_{ui}|n_{ui} \sim \mathrm{Poisson}(n_{ui}\kappa_2(-\log(1 - e^\theta)))$. This leads to the following update:

$$\kappa_2 \leftarrow \frac{1}{-\log(1 - e^\theta)} \frac{\sum_{ui}\langle m_{ui}\rangle_q}{\sum_{ui}\langle n_{ui}\rangle_q}, \quad (17)$$

where $\langle m_{ui}\rangle_q = \sum_{n=1}^{y_{ui}} n\kappa_2\big(\Psi(y_{ui} - n + n\kappa_2) - \Psi(n\kappa_2)\big) q(n_{ui} = n)$.

6 EXPERIMENTAL RESULTS

Datasets. We consider the following datasets, whose structure is summarized in Table 2.

• The Taste Profile dataset (Bertin-Mahieux et al., 2011) provided by The Echo Nest contains the listening history of users. The data collected are the number of time the users listened to songs. We pre-process a subset of the data as in (Liang et al., 2016), keeping only users and songs that have more than 20 interactions. The histogram of the listening counts is displayed on Figure 2.

• The Last.fm dataset (Celma, 2010) contains the listening history of users with additional timestamps information. We select play counts of the year 2008 and apply the same pre-processing as with the Taste Profile dataset.

• The NIPS dataset (Perrone et al., 2016) contains bag-of-words representations of conference papers published

Table 2: Datasets structure after pre-processing.

	Taste Profile	NIPS	Last.fm
# rows	$16,301$	$5,811$	781
# columns	$12,118$	$11,463$	$11,172$
# non-zeros	$1,176,086$	$4,033,830$	$402,058$
% non-zeros	0.60%	6.06%	4.61%

in the NIPS conference from 1987 to 2015. We make an analogy between "users who listened to songs" and "documents written with words". The goal here is to recommend unused words to the author of a paper.

Experimental setup. Each dataset is split into a train set $\mathbf{Y}^{\text{train}}$ containing 80% of the non-zero of the original dataset and a test set $\mathbf{Y}^{\text{test}}$ containing the remaining 20% (these values being set to 0 in the train set). All the compared algorithms are trained with the train set and provide a recommendation list for each user. These lists are evaluated with the test set.

For each user, we recommend an ordered list of m songs he/she never listened to, based on $\mathbf{Y}^{\text{train}}$. The songs in this list are sorted w.r.t. the prediction score defined by: $s_{ui} = \sum_{k=1}^{K} \langle w_{uk} \rangle_q \langle h_{ik} \rangle_q$. Note that, in dcPF, the expected number of listening sessions is equal to $[\mathbf{W}\mathbf{H}^T]_{ui}$, expressing long-term preference (see Section 3.1).

The quality of the proposed list is measured by the normalized discounted cumulative gain (NDCG) score. NDCG is a metric often used in information retrieval to evaluate lists of ranked items. It is calculated as follows:

$$\text{DCG}_u = \sum_{i=1}^{m} \frac{\text{rel}(u,i)}{\log_2(i+1)}, \ \ \text{NDCG}_u = \frac{\text{DCG}_u}{\text{IDCG}_u},$$

where DCG_u is the discounted cumulative gain and $\text{rel}(u,i)$ is the relevance to the ground-truth of the ith item of the proposed list. IDCG_u is the ideal DCG, i.e., the best DCG score that can be obtained. Therefore $\text{NDCG}_u \in [0,1]$ with $\text{NDCG}_u = 1$ corresponding to the perfect recommended list. For the relevance to the ground-truth, we propose to account for the values in the test set above a fixed threshold: $\text{rel}(u,i) = \mathbb{1}[y_{ui}^{\text{test}} > s]$. As mentioned in (Hu et al., 2008), small listening counts reflect a preference with little confidence. For example, a user can listen to a song by pure curiosity. Therefore, this threshold leads to a more robust NDCG metric. We denote by $\text{NDCG}s$ the NDCG with the threshold s. If $s = 0$, we recover the classic $\text{NDCG}0$ metric for binary data. Otherwise, for $s \geq 1$, $\text{NDCG}s$ only considers the songs which have been listened to at least s times and ignores the others. Other metrics such as precision and recall lead to similar conclusions than $\text{NDCG}0$ and will not be displayed in the following.

Compared methods. For the three datasets, we compare dcPF with its limit cases: PF performed on raw (PFraw) or on binarized data (PFbin). PF is known to achieve good performance in recommendation tasks (Gopalan et al., 2015). For the Taste Profile dataset, we considered the hyper-parameters $\alpha^W = \alpha^H$ among $\{0.1, 0.3, 1\}$ and K among $\{50, 100, 200\}$, and selected the values $\alpha^W = \alpha^H = 0.3$ and $K = 100$ which gave the best NDCG0 for PFbin. For the NIPS and Last.fm datasets, which are smaller than the Taste Profile dataset, we only considered $\alpha^W = \alpha^H = 0.3$ and $K = 50$. The stopping criterion of the algorithms is set to $\tau = 10^{-5}$. Evaluation is done using a ranked list of $m = 100$ items. For all the experiments, algorithms are run five times from random initializations.[4]

Prediction results. We start by discussing results with the Taste Profile dataset, reported in Table 3. A general observation is that dcPF gives better results than the two baselines PFraw and PFbin for all four metrics and every element distribution, with the exception of ZTP in the case of NDCG0. PFbin returns better scores than PFraw up to $s = 5$. This confirms the usefulness of the binarization stage when using PF, but only up to a certain threshold s (this is because PFbin does not fully exploit the non-zero values in the original raw data). The performance gap between dcPF and PFbin increases with the threshold s. On the two other datasets (NIPS and Last.fm), Table 4 shows that dcPF outperforms the baseline methods for all element distributions. Note that for the NIPS dataset, a somehow different context from song recommendation, PFraw is effective and performs better than PFbin as soon as the threshold s is larger than one. From both Tables 3 and 4 we conclude that the proposed shifted NB element distribution is a good compromise overall.

Natural parameter estimation. As explained in Section 3.3, estimation of the natural parameter tells us about the level of scale information exploited by dcPF. Table 3 shows that dcPF indeed offers a valuable trade-off between PFraw and PFbin, because the estimated parameter θ lies in between the two limit cases θ^{raw} and θ^{bin}. To assess the quality of the estimation procedures for θ in Log, ZTP and Geo described in Section 3.3 (plainly referred as VBEM), Table 3 also displays evaluation metrics obtained with a grid-search. More precisely, we use $\theta = \log p$ which maximizes NDCG5 from a set of pre-specified values. For Log and Geo, p is searched between 0 and 1 with a step of 0.1. For ZTP, p is searched in $\{0, 0.1, 0.5, 1, 2, 10, 100, +\infty\}$. It appears that VBEM slightly over-estimates the optimal value (in terms of

---

[4]Algorithms and experiments are available on github: `https://github.com/Oligou/dcPF`.

Table 3: Recommendation performance of dcPF and PF using the Taste Profile dataset. Italic: scores of dcPF when using a grid-search for θ, see text for details. Bold: two best NDCG scores (grid-search excluded).

Model	Est.	$p = e^\theta$	κ	NDCG0	NDCG1	NDCG2	NDCG5
Log	VBEM	0.803	1	**0.200** $(\pm\mathbf{3.0\,10^{-3}})$	**0.182** $(\pm\mathbf{2.3\,10^{-3}})$	**0.166** $(\pm\mathbf{2.0\,10^{-3}})$	0.147 $(\pm 1.5\,10^{-3})$
	Grid	*0.3*	*1*	*0.200* $(\pm 4.1\,10^{-3})$	*0.186* $(\pm 3.9\,10^{-3})$	*0.173* $(\pm 3.7\,10^{-3})$	*0.158* $(\pm 3.7\,10^{-3})$
ZTP	VBEM	1.950	1	0.192 $(\pm 4.1\,10^{-3})$	0.178 $(\pm 3.7\,10^{-3})$	**0.167** $(\pm\mathbf{3.6\,10^{-3}})$	**0.156** $(\pm\mathbf{3.8\,10^{-3}})$
	Grid	*1*	*1*	*0.190* $(\pm 3.5\,10^{-3})$	*0.178* $(\pm 3.0\,10^{-3})$	*0.168* $(\pm 3.1\,10^{-3})$	*0.158* $(\pm 3.3\,10^{-3})$
Geo	VBEM	0.600	1	0.199 $(\pm 2.3\,10^{-3})$	**0.182** $(\pm\mathbf{1.8\,10^{-3}})$	**0.167** $(\pm\mathbf{1.8\,10^{-3}})$	0.150 $(\pm 1.2\,10^{-3})$
	Grid	*0.3*	*1*	*0.199* $(\pm 4.9\,10^{-3})$	*0.185* $(\pm 4.6\,10^{-3})$	*0.172* $(\pm 4.2\,10^{-3})$	*0.159* $(\pm 4.0\,10^{-3})$
Sh. NB	VBEM	0.873	$(1, 0.2)^T$	**0.201** $(\pm\mathbf{3.1\,10^{-3}})$	**0.183** $(\pm\mathbf{2.5\,10^{-3}})$	**0.166** $(\pm\mathbf{2.2\,10^{-3}})$	0.147 $(\pm 1.5\,10^{-3})$
PFraw	.	.	.	0.156 $(\pm 3.0\,10^{-3})$	0.155 $(\pm 3.3\,10^{-3})$	0.150 $(\pm 3.5\,10^{-3})$	0.144 $(\pm 5.3\,10^{-3})$
PFbin	.	.	.	0.197 $(\pm 2.1\,10^{-3})$	0.177 $(\pm 1.5\,10^{-3})$	0.160 $(\pm 1.5\,10^{-3})$	0.140 $(\pm 1.3\,10^{-3})$

Table 4: Performance with NIPS and Last.fm datasets.

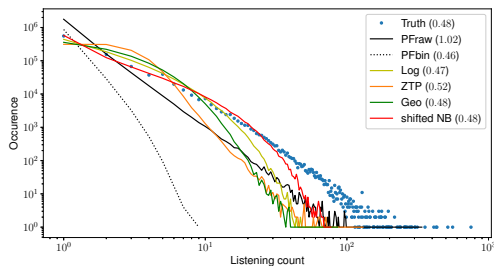| Model	NIPS		Last.fm	
	NDCG0	NDCG1	NDCG0	NDCG1
Log	**0.394**	**0.430**	0.142	0.129
ZTP	0.381	0.422	0.122	0.113
Geo	0.390	0.429	0.139	0.128
Sh. NB	**0.396**	**0.431**	**0.143**	**0.130**
PFraw	0.358	0.405	0.091	0.088
PFbin	0.378	0.392	0.122	0.108

Figure 2: PPC of the distribution of the non-zero values in the Taste Profile dataset. The blue points (Truth) represents the histogram of the non-zero values in the train set. The colored curves represent the simulated histograms obtained from the different inferred dcPF or PF models. The percentages of non-zero values are written in parentheses.

NDCG5) of the natural parameter, but remains a very robust procedure.

Posterior predictive check (PPC). We provide a PPC of the distribution of the listening counts in the Taste Profile dataset (see Figure 2). A PPC consists in simulating a new dataset $\mathbf{Y}^{\mathrm{PPC}}$ from the fitted model (for dcPF, we simulate from the generative process described in Section 3.1 with latent variables $\mathbf{W}$, $\mathbf{H}$ and parameters inferred in Section 6). Then, we compare the histogram of the values of $\mathbf{Y}^{\mathrm{train}}$ and $\mathbf{Y}^{\mathrm{PPC}}$. The PPC of the two limit cases is very instructive. PFraw tries to fit the long tail of the data, but, by doing so, destroys the representation of the zero values (1.02% of non-zero values versus 0.48% in the real

dataset). It can explain the disappointing performances of PFraw for NDCG0. On the contrary, PFbin better fits the sparsity of the data but is not able to describe large counts. In both cases, PF struggles to properly weigh the influence of large counts compared to low counts. Comparatively, dcPF proposes a smoother weighting between large and low values. dcPF respects both the sparsity and the long tail of the data for the four element distributions. ZTP seems to over-estimate the influence of medium counts (from 1 to 5), whereas shifted NB has the best fit to the histogram. We observe that regardless of the model, explaining the large values (> 100) remains difficult, however we may consider that after a certain threshold the counts do not contain useful information.

7 CONCLUSION

In this paper, we described new contributions to cPF for discrete data. As compared to PF, we showed that dcPF offers more flexibility to model long-tailed data. Inference remains scalable thanks to modeling of the non-zero values only. Numerical experiments confirmed that our adaptive VBEM algorithm efficiently exploits raw data, leading to better recommendation scores when compared to the two limit cases (PF on raw and binarized data). Among the four element distributions presented and experimented in this work, the proposed shifted NB prove particularly efficient thanks to its additional parameter, and often led to the best recommendation scores.

Based on this work, a number of exciting perspectives can be considered, such as investigating more complex element distributions to better fit the extreme observations or to address other forms of data. For instance, it would be of great interest to adapt the model for bounded data such as ratings, which are widespread in CF but cannot be processed with dcPF in its current form.

References

M. E. Basbug and B. E. Engelhardt. 2016. Hierarchical compound Poisson factorization. In *Proc. International Conference on Machine Learning (ICML)*.

J. Bennett and S. Lanning. 2007. The Netflix Prize. In *Proc. KDD Cup and Workshop*.

T. Bertin-Mahieux, D. PW Ellis, B. Whitman, and P. Lamere. 2011. The Million Song Dataset. In *Proc. International Society for Music Information Retrieval (ISMIR)*, Vol. 2.

C. M. Bishop. 2006. *Pattern recognition and machine learning*. Springer.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. 2017. Variational inference: a review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

D. M. Blei, Andrew Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

W. E. Bleick and P. CC Wang. 1974. Asymptotics of Stirling numbers of the second kind. In *Proc. of the American Mathematical Society*, Vol. 42.

J. Canny. 2004. GaP: a factor model for discrete data. In *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*.

O. Celma. 2010. Music recommendation. In *Music recommendation and discovery*. Springer, 43–85.

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

N. Du, Y. Wang, N. He, J. Sun, and L. Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems (NIPS)*.

R. A. Fisher, A. S. Corbet, and C. B. Williams. 1943. The relation between the number of species and the number of individuals in a random sample of an animal population. *The Journal of Animal Ecology* (1943), 42–58.

C. Févotte and J. Idier. 2011. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural computation* 23, 9 (2011), 2421–2456.

P. Gopalan, J. M. Hofman, and D. M. Blei. 2015. Scalable recommendation with hierarchical Poisson factorization. In *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*.

P. K. Gopalan, L. Charlin, and D. M. Blei. 2014. Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems (NIPS)*.

S. Hosseini, A. Khodadadi, K. Alizadeh, A. Arabzadeh, M. Farajtabar, H. Zha, and H. R. Rabiee. 2018. Recurrent Poisson factorization for temporal recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018).

Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proc. IEEE International Conference on Data Mining (ICDM)*.

N. L. Johnson, A. W. Kemp, and S. Kotz. 2005. *Univariate discrete distributions*. Vol. 444. John Wiley & Sons.

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

B. Jørgensen. 1986. Some properties of exponential dispersion models. *Scandinavian Journal of Statistics* (1986), 187–197.

B. Jørgensen. 1987. Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Methodological)* (1987), 127–162.

A. Khodadadi, S. A. Hosseini, E. Tavakoli, and Hamid R. Rabiee. 2018. Continuous-time user modeling in the presence of badges: a probabilistic approach. *ACM Transactions on Knowledge Discovery from Data* 12, 3 (2018), 37.

J. Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery* (2003).

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

J. F. Lawless. 1987. Negative binomial and mixed Poisson regression. *Canadian Journal of Statistics* 15, 3 (1987), 209–225.

D. D. Lee and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

D. D. Lee and H. S. Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*.

D. Liang, L. Charlin, J. McInerney, and D. M. Blei. 2016. Modeling user exposure in recommendation. In *Proc. International Conference on World Wide Web (WWW)*.

D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proc. International Conference on World Wide Web (WWW)*.

H. Lu, W. Niu, and J. Caverlee. 2018. Learning geo-social user topical profiles with Bayesian hierarchical user

factorization. In *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*.

H. Ma, C. Liu, I. King, and M. R. Lyu. 2011. Probabilistic factor models for web site recommendation. In *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*.

R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. 2008. One-class collaborative filtering. In *Proc. IEEE International Conference on Data Mining (ICDM)*.

V. Perrone, P. A. Jenkins, D. Spano, and Y. W. Teh. 2016. Poisson random fields for dynamic feature models. *arXiv preprint arXiv:1611.07460* (2016).

M. H. Quenouille. 1949. A relation between the logarithmic, Poisson, and negative binomial series. *Biometrics* 5, 2 (1949), 162–164.

R. Ranganath, L. Tang, L. Charlin, and D. M. Blei. 2015. Deep exponential families. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*.

J. Riordan. 2012. *Introduction to combinatorial analysis*. Courier Corporation.

C. Robert. 2007. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media.

A. Salah and H. W. Lauw. 2018. Probabilistic collaborative representation learning for personalized item recommendation. In *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*.

A. Schein, H. Wallach, and M. Zhou. 2016. Poisson-gamma dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*.

U. Simsekli, A. T. Cemgil, and Y. K. Yilmaz. 2013. Learning the beta-divergence in Tweedie compound Poisson matrix factorization models. In *Proc. International Conference on Machine Learning (ICML)*.

C. Wang, D. M Blei, et al. 2018. A general method for robust Bayesian modeling. *Bayesian Analysis* 13, 4 (2018), 1163–1191.

Y. K. Yılmaz and A. T. Cemgil. 2012. Alpha/beta divergences and Tweedie models. *arXiv preprint arXiv:1209.4280* 1050 (2012), 19.

M. Zhou. 2018. Nonparametric Bayesian negative binomial factor analysis. *Bayesian Analysis* 13, 4 (2018), 1065–1093.

M. Zhou and L. Carin. 2015. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 307–320.

M. Zhou, Y. Cong, and B. Chen. 2015. The Poisson gamma belief network. In *Advances in Neural Information Processing Systems (NIPS)*.