Bayesian mean-parameterized nonnegative binary matrix factorization

Alberto Lumbreras, Louis Filstroff, Cédric Févotte

Bayesian Mean-parameterized Nonnegative Binary Matrix Factorization

Alberto Lumbreras[1], Louis Filstroff[2], and Cédric Févotte[2]

[1]Criteo AI Lab, France
[2]IRIT, Université de Toulouse, CNRS, France

June 23, 2020

Abstract

Binary data matrices can represent many types of data such as social networks, votes, or gene expression. In some cases, the analysis of binary matrices can be tackled with nonnegative matrix factorization (NMF), where the observed data matrix is approximated by the product of two smaller nonnegative matrices. In this context, probabilistic NMF assumes a generative model where the data is usually Bernoulli-distributed. Often, a link function is used to map the factorization to the $[0, 1]$ range, ensuring a valid Bernoulli mean parameter. However, link functions have the potential disadvantage to lead to uninterpretable models. Mean-parameterized NMF, on the contrary, overcomes this problem. We propose a unified framework for Bayesian mean-parameterized nonnegative binary matrix factorization models (NBMF). We analyze three models which correspond to three possible constraints that respect the mean-parameterization without the need for link functions. Furthermore, we derive a novel collapsed Gibbs sampler and a collapsed variational algorithm to infer the posterior distribution of the factors. Next, we extend the proposed models to a nonparametric setting where the number of used latent dimensions is automatically driven by the observed data. We analyze the performance of our NBMF methods in multiple datasets for different tasks such as dictionary learning and prediction of missing data. Experiments show that our methods provide similar or superior results than the state of the art, while automatically detecting the number of relevant components.

1 Introduction

Nonnegative matrix factorization (NMF) is a family of methods that approximate a nonnegative matrix $\mathbf{V}$ of size $F \times N$ as the product of two nonnegative matrices,

$$\mathbf{V} \approx \mathbf{WH}, \tag{1}$$

where $\mathbf{W}$ has size $F \times K$, and $\mathbf{H}$ has size $K \times N$, often referred to as the *dictionary* and the *activation matrix*, respectively. K is usually chosen such that $FK + KN \ll FN$, hence reducing the data dimension.

Such an approximation is often sought after by minimizing a measure of fit between the observed data $\mathbf{V}$ and its factorized approximation $\mathbf{WH}$, i.e.,

$$\mathbf{W}, \mathbf{H} = \underset{\mathbf{W},\mathbf{H}}{\arg\min}\, D(\mathbf{V}|\mathbf{WH}) \quad \text{s.t} \quad \mathbf{W} \geq 0, \quad \mathbf{H} \geq 0, \tag{2}$$

where D denotes the cost function, and where the notation $\mathbf{A} \geq 0$ denotes nonnegativity of the entries of $\mathbf{A}$. Typical cost functions include the squared Euclidean distance and the generalized Kullback-Leiber divergence (Lee and Seung, 2001), the α-divergence (Cichocki et al., 2008) or the β-divergence (Févotte and Idier, 2011). Most of these cost functions underlie a probabilistic model for the data, such that minimization of the cost function is equivalent to joint maximum likelihood estimation of the factors (Singh and Gordon, 2008), i.e.,

$$\arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}|\mathbf{WH}) = \arg\max_{\mathbf{W},\mathbf{H}} p(\mathbf{V}|\mathbf{W},\mathbf{H}), \tag{3}$$

where p is a probability distribution. As such, so-called *Bayesian NMF* can be considered, where the factors $\mathbf{W}$ and $\mathbf{H}$ are assumed to be random variables with prior distributions, and inference is based on their posterior distribution, i.e.,

$$p(\mathbf{W},\mathbf{H}|\mathbf{V}) = p(\mathbf{V}|\mathbf{W},\mathbf{H})p(\mathbf{W},\mathbf{H})/p(\mathbf{V}). \tag{4}$$

This has notably been addressed for different models such as Poisson (Cemgil, 2009), additive Gaussian (Schmidt et al., 2009; Alquier and Guedj, 2017), or multiplicative Exponential (Hoffman et al., 2010).

In this paper, we are interested in Bayesian NMF for binary data matrices. Binary matrices may represent a large variety of data such as social networks, voting data, gene expression data, or binary images. As we shall see in Section 2, a common practice is to consider the model

$$p(\mathbf{V}|\mathbf{W},\mathbf{H}) = \prod_{f,n} \text{Bernoulli}\left(v_{fn}|\phi([\mathbf{WH}]_{fn})\right), \tag{5}$$

where ϕ is a link function that maps the factorization $\mathbf{WH}$ to the $[0,1]$ range.[1] Although link functions are convenient since they allow the factors to be unconstrained, and sometimes result in tractable problems, they sacrifice the mean-parameterization of the Bernoulli likelihood (i.e. $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = \phi(\mathbf{WH})$ instead of $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = \mathbf{WH}$).

Mean-parameterized nonnegative binary matrix factorization (NBMF), however, does not rely on a link function —or equivalently, considers $\phi(\mathbf{WH}) = \mathbf{WH}$— and assumes the likelihood of the data to be

$$p(\mathbf{V}|\mathbf{W},\mathbf{H}) = \prod_{f,n} \text{Bernoulli}(v_{fn}|[\mathbf{WH}]_{fn}),$$

which implies $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = \mathbf{WH}$. Mean-parameterization is an interesting property of a model because it makes the decomposition easy to interpret. Besides, in a Bernoulli likelihood, the product $\mathbf{WH}$ —and, in this paper, the individual factors as well— can be interpreted as probabilities. An additional advantage of dealing with probabilities is that they lay in a linear, continuous space, where we can apply off-the-shelf clustering methods over the latent factors. For instance, in recommender systems, we may want to cluster users by latent musical preferences, or by the latent type of product they buy. Or we may want a user to see only those categories with a probability higher than some threshold.

Our contributions in this paper are the following:

(a) we present a unified framework for three Bayesian mean-parameterized NBMF models that place three possible constraints on the factors;

(b) we derive a collapsed Gibbs sampler as well as collapsed variational inference algorithms, which have never been considered for these models;

[1] Distributions used throughout the article are formally defined in Appendix A.

(c) we discuss the extension of the models to a nonparametric setting —where the number of latent components does not need to be fixed a priori— and propose an approximation that shows excellent results with real data.

We test the performance of the models for different tasks in multiple datasets and show that our models give similar or superior results to the state of the art, while automatically detecting the number of relevant components. The datasets, the algorithms, and scripts to replicate all the reported results are available through an R package. [2]

2 Related work

2.1 Logistic PCA family

One of the earliest probabilistic approaches to model binary data matrices comes from PCA-related methods. The reformulation of PCA as a probabilistic generative model with a Gaussian likelihood (Sammel et al., 1997; Tipping and Bishop, 1999) opened the door to considering other likelihoods such as Bernoulli models, which are more appropriate for binary observations. We refer to it as logistic PCA. The maximum likelihood estimator in the model is given by

$$v_{fn} \sim \text{Bernoulli}\left(\sigma\left([\mathbf{WH}]_{fn}\right)\right), \tag{6}$$

where σ is the logistic function $\sigma(x) = 1/(1+e^{-x})$. Note that in this model the expectation is a non-linear transformation of the factors, such that $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = \sigma(\mathbf{WH})$.

There are multiple maximum likelihood estimation algorithms for logistic PCA. For instance, while Sammel et al. (1997) use a Monte-Carlo Expectation Minimization (MC-EM) algorithm, Tipping (1999) derives a faster variational EM (vEM) algorithm. Collins et al. (2002) generalize probabilistic PCA to the exponential family and propose a general algorithm that exploits the duality between likelihoods in the exponential family and Bregman divergences. Later, Schein et al. (2003) improved the algorithm of Collins et al. (2002), thanks to the optimization of a tight upper bound by Alternate Least Squares (ALS). Finally, note that inference could also be tackled with Polya-Gamma data augmentation schemes Polson et al. (2013).

Other models similar to logistic PCA have been proposed with various priors or constraints over the factors. Some examples are Hernandez-Lobato et al. (2014), where the factors are given Gaussian priors, Tomé et al. (2013), which allows one factor to have negative values, and Larsen and Clemmensen (2015), where both factors are nonnegative. Meeds et al. (2007) consider the same logistic link function but a three factor decomposition $\sigma(\mathbf{WXH})$, where $\mathbf{W}$ and $\mathbf{H}$ are binary factors that represent cluster assignments, and $\mathbf{X}$ is a real-valued matrix that encodes the relations between the clusters. The expectation in these models is always $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = \sigma(\mathbf{WH})$ or $\mathbb{E}[\mathbf{V}|\mathbf{WXH}] = \sigma(\mathbf{WXH})$.

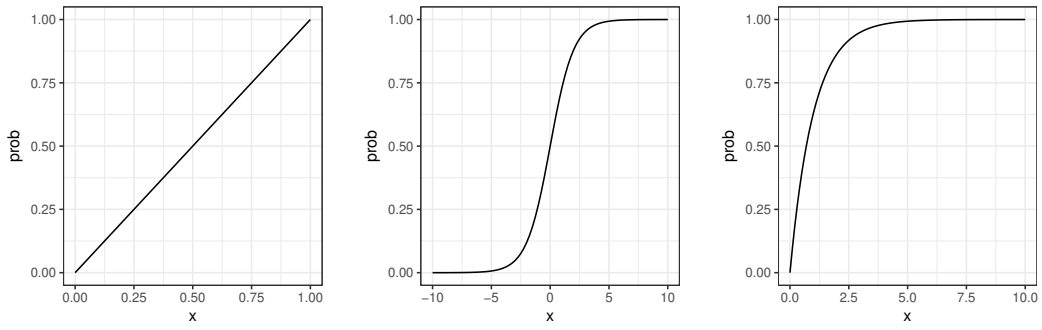
2.2 Poisson matrix factorization

For practical reasons, some works have considered Poisson matrix factorization (PMF) techniques for binary data. In this case the binary nature of the data is ignored and a Poisson likelihood is considered:

$$v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}). \tag{7}$$

Different flavors of PMF have been proposed, in frequentists or Bayesian settings, and can be found, for example, in Lee and Seung (2001); Canny (2004); Cemgil (2009); Zhou et al. (2012);

[2] https://github.com/alumbreras/NBMF



(a) Identity **(b)** Logistic **(c)** Bernoulli-Poisson

Figure 1 Three possible link functions (identity induces mean-parameterization).

Gopalan et al. (2014, 2015). An advantage of PMF is that it is mean-parameterized, i.e., $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = \mathbf{WH}$. Another useful advantage is that inference algorithms need only iterate over non-zero values, which makes them very efficient for sparse matrices. In our case, a significant disadvantage is that it assigns non-zero probabilities to impossible observations ($v_{fn} > 1$).

As we discussed above, a more reasonable choice consists in replacing the Poisson distribution with a Bernoulli distribution, possibly using some link function that maps the parameter into a $[0, 1]$ range, ensuring a valid Bernoulli parameter. Unfortunately, unlike Poisson models, zeroes and ones under Bernoulli likelihoods do not represent counts but classes —a zero can be considered as a *no*, while a one can be considered as a *yes*— and the algorithms need to iterate over all the elements of the observation matrix. To bypass this, Zhou (2015); Zhou et al. (2016) proposed using the alternative link function $f(x) = 1 - e^{-x}$, coined Bernoulli-Poisson, such that

$$v_{fn} \sim \text{Bernoulli}\left(f\left([\mathbf{WH}]_{fn}\right)\right). \tag{8}$$

Thanks to the new link function, the model can be "augmented" to a Poisson model by introducing latent variables c_{fn} such that

$$c_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}) \tag{9}$$

$$v_{fn} = \mathbb{1}[c_{fn} \geq 1], \tag{10}$$

where $\mathbb{1}$ is the set indicator function. By placing conjugate Gamma priors over the factors, posteriors can be obtained by Gibbs sampling. Expectation in this model is $\mathbb{E}[\mathbf{V}|\mathbf{WH}] = f(\mathbf{WH})$. Figure 1 shows the logistic and Bernoulli-Poisson functions. Note that each link function has a different input domain, leading to different priors or constraints over the factors.

2.3 Bernoulli mean-parameterized matrix factorization

None of the above methods offers mean-parameterization *and* assumes a Bernoulli distribution over the data. The family of *mean-parameterized Bernoulli models* is the basis of the models presented in this paper. These models have first been introduced in Kabán and Bingham (2008) (binary ICA) and Bingham et al. (2009) (Aspect Bernoulli model) where the constraints $\sum_k w_{fk} = 1, h_{kn} \in [0, 1]$, or vice versa, are imposed on the factors. The trick is that these constraints induce convex combinations of binary elements such that $\sum_k w_{fk} h_{kn} \in [0, 1]$, which gives a valid Bernoulli parameter. In Bingham et al. (2009) the constraints are imposed explicitly, and the maximum likelihood estimator is computed by using EM with an augmented version

Table 1 Bernoulli matrix factorization methods considered in the literature. Bernoulli, Gamma, Dirichlet distributions are denoted as Ber, Ga, and Dir, respectively. Gradient refers to Gradient-based optimization.

Reference	Likelihood	Prior / Constr.	Estimation
Sammel et al. (1997)	$\text{Ber}(\sigma([\mathbf{WH}]_{fn}))$	$w_{fk} \sim \text{Normal}(\cdot)$ $h_{kn} \in \mathbb{R}$	MC-EM
Tipping (1999)	$\text{Ber}(\sigma([\mathbf{WH}]_{fn}))$	$w_{fk} \sim \text{Normal}(\cdot)$ $h_{kn} \in \mathbb{R}$	vEM
Collins et al. (2002)	$\text{Ber}(\sigma([\mathbf{WH}]_{fn}))$	$w_{fk} \in \mathbb{R}$ $h_{kn} \in \mathbb{R}$	Gradient
Schein et al. (2003)	$\text{Ber}(\sigma([\mathbf{WH}]_{fn}))$	$w_{fk} \in \mathbb{R}$ $h_{kn} \in \mathbb{R}$	ALS
Meeds et al. (2007)	$\text{Ber}(\sigma([\mathbf{WXH}]_{fn}))$	$w_{fk} \sim \text{Ber}(\cdot)$ $h_{kn} \sim \text{Ber}(\cdot)$	Gibbs
Kabán and Bingham (2008)	$\text{Ber}([\mathbf{WH}]_{fn})$	$w_{kn} \sim \text{Beta}(\cdot)$ $\mathbf{h}_n \sim \text{Dir}(\cdot)$	VB
Bingham et al. (2009)	$\text{Ber}([\mathbf{WH}]_{fn})$	$\sum_k w_{fk} = 1$ $h_{kn} \in [0, 1]$	EM
Tomé et al. (2013)	$\text{Ber}(\sigma([\mathbf{WH}]_{fn}))$	$w_{fk} \in \mathbb{R}_+$ $h_{kn} \in \mathbb{R}$	Gradient
Larsen and Clemmensen (2015)	$\text{Ber}(\sigma([\mathbf{WH}]_{fn}))$	$w_{fk} \in \mathbb{R}_+$ $h_{kn} \in \mathbb{R}_+$	Gradient
Zhou (2015)	$\text{Ber}(f([\mathbf{WH}]_{fn}))$	$w_{fk} \sim \text{Ga}(\cdot)$ $h_{kn} \sim \text{Ga}(\cdot)$	Gibbs

of the model. In Kabán and Bingham (2008) the constraint is imposed through Dirichlet and Beta priors over the factors, and Variational Bayes (VB) estimation of their posteriors is derived exploiting a similar augmentation scheme.

Table 1 presents a summary of the methods presented in Sections 2.1-2.3 that use a Bernoulli likelihood.

2.4 Others

Some models have also been proposed to find binary decompositions, that is, matrix factorizations where $\mathbf{W}$ and $\mathbf{H}$ contain binary elements. For instance, Zhang et al. (2009) aim to minimize a Euclidean distance or, equivalently, maximize a Gaussian likelihood under the binary constraint. Similarly, the "discrete basis problem" of Miettinen et al. (2008) aims to minimize a $L1$-norm under the same constraint. In Slawski et al. (2013), an algorithm is proposed to retrieve the exact factorization when one of the factors is constrained to be binary, and the other one to be stochastic, i.e., $\sum_k h_{kn} = 1$. More recently Rukat et al. (2017) proposed a Bayesian model for the Boolean matrix factorization problem, where $\mathbf{WH}$ is a Boolean product. Çapan et al. (2018) proposed sum-conditioned Poisson factorization models that apply to binary data.

In the general model defined by Eq. (5), we are essentially assuming that $\mathbf{V} \approx \phi(\mathbf{WH})$. This can be seen as a one-layer generative network with input $\mathbf{H}$, weight $\mathbf{W}$ and non-linearity $\phi(\cdot)$. As such it is possible to conceive more general models by stacking various layers or modeling $\mathbf{W}$ and $\mathbf{H}$ as the outputs of deep networks themselves. In the context of recommendation systems, where binary matrices can represent either binary ratings or implicit feedback, this has been for

example considered in He et al. (2017); Xue et al. (2017).

3 Mean-parameterized Bernoulli models

3.1 Models

Let us consider a mean-parameterized Bernoulli model for an observed binary matrix $\mathbf{V}$ and two latent factors $\mathbf{W}$, $\mathbf{H}$:

$$v_{fn} \sim \text{Bernoulli}([\mathbf{WH}]_{fn}). \tag{11}$$

To guarantee valid Bernoulli parameters, we can impose three possible sets of constraints on the factors such that $\sum_k w_{fk} h_{kn} \in [0, 1]$:

(c1)	(c2)	(c3)
$h_{kn} \in [0, 1]$	$\sum_k h_{kn} = 1$	$\sum_k h_{kn} = 1$
$\sum_k w_{fk} = 1$	$w_{fk} \in [0, 1]$	$\sum_k w_{fk} = 1$

In a Bayesian setting, we may place Beta and Dirichlet priors over the factors to respect these constraints:

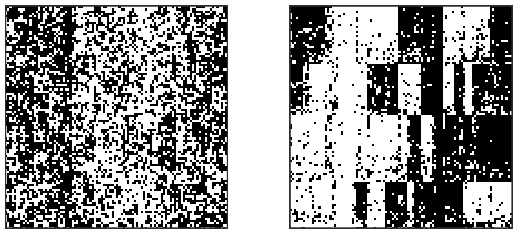
`Beta-Dir` (c1)	`Dir-Beta` (c2)	`Dir-Dir` (c3)
$h_{kn} \sim \text{Beta}(\alpha_k, \beta_k)$	$\mathbf{h}_n \sim \text{Dirichlet}(\boldsymbol{\eta})$	$\mathbf{h}_n \sim \text{Dirichlet}(\boldsymbol{\eta})$
$\mathbf{w}_f \sim \text{Dirichlet}(\boldsymbol{\gamma})$	$w_{fk} \sim \text{Beta}(\alpha_k, \beta_k)$	$\mathbf{w}_f \sim \text{Dirichlet}(\boldsymbol{\gamma})$

where $\mathbf{h}_n$ denotes the n-th column of the matrix $\mathbf{H}$, and $\mathbf{w}_f$ denotes the f-th row of the matrix $\mathbf{W}$. The Beta parameters are positive real numbers $\alpha_k, \beta_k \in \mathbb{R}_{++}$ and the Dirichlet parameters are K-dimensional vectors of positive real numbers $\boldsymbol{\gamma}, \boldsymbol{\eta} \in \mathbb{R}^K_{++}$.

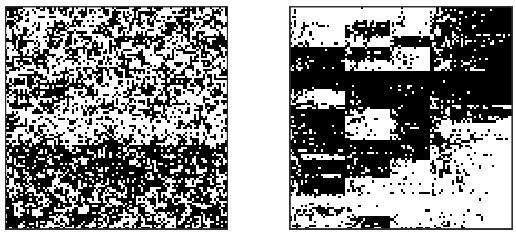
Note that each element w_{fk} and h_{kn} can be interpreted as a probability. We can either merely impose that the elements of a row $\mathbf{w}_f$ or a column $\mathbf{h}_n$ lie between 0 and 1, or, more strongly, that they sum up to one. This implies a difference in modeling. On the one hand, imposing that the elements lie between 0 and 1 induce non-exclusive components. On the other hand, the sum-to-one constraint induces exclusive components, i.e., the more likely a component is, the less likely are the others. Fig. 2 displays simulated matrices generated from each of the models.

The first two models, `Beta-Dir` and `Dir-Beta`, are symmetric. Indeed, estimating $\mathbf{W}$ (resp., $\mathbf{H}$) in one model is equivalent to estimating $\mathbf{H}$ (resp., $\mathbf{W}$) in the other model after transposing the matrix $\mathbf{V}$. As such, in the rest of the paper, we will only consider the `Beta-Dir` model and the `Dir-Dir` model.

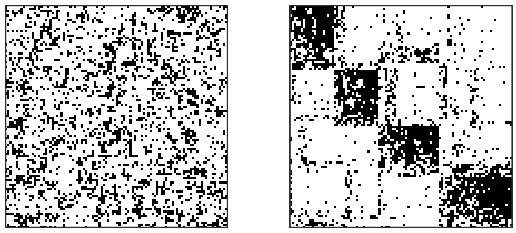
The Aspect Bernoulli model of Bingham et al. (2009) is built over Eq. (11), and considers that the factors $\mathbf{W}$ and $\mathbf{H}$ are deterministic parameters which satisfy the constraint (c2). The factors are estimated by maximum likelihood with EM. The binary ICA of Kabán and Bingham (2008) corresponds to the `Beta-Dir` model, and inference is performed with VB. In the following sections, we present inference methods for the posterior distributions of $\mathbf{W}$ and $\mathbf{H}$ in the `Beta-Dir` and `Dir-Dir` models. Because these distributions are intractable, we propose novel collapsed Gibbs sampling and collapsed variational inference strategies. We also derive a nonparametric approximation where the number of latent dimensions K does not need to be fixed a priori.



(a) `Beta-Dir`



(b) `Dir-Beta`



(c) `Dir-Dir`

Figure 2 Synthetic 100×100 matrices drawn from the three generative models with $K = 4$. Matrices on the left are generated with $\alpha_k = \beta_k = \gamma_k = \eta_k = 1$. Matrices on the right are generated with $\alpha_k = \beta_k = \gamma_k = \eta_k = 0.1$. For better visualization, rows and columns are re-ordered according to complete linkage clustering (Sørensen, 1948) using the `hclust` function in R (R Core Team, 2017).

3.2 Connections with latent Dirichlet allocation (LDA).

The proposed models have connection with topic models, in particular with LDA (Blei et al., 2003), described by

$$\mathbf{h}_n \sim \text{Dirichlet}(\boldsymbol{\gamma}) \tag{12}$$

$$\mathbf{v}_n | \mathbf{h}_n \sim \text{Multinomial}\,(L_n, \mathbf{W}\mathbf{h}_n). \tag{13}$$

Here, $\mathbf{V}$ is the so-called "bag-of-words" representation of a corpus of documents, i.e., v_{fn} represents the number of occurrences of word f in document n. L_n is the total number of words in document n, i.e., $L_n = \sum_f v_{fn}$. The columns of $\mathbf{W}$ are assumed to sum to 1, such that w_{fk} represents the probability of word f in topic k, and h_{kn} represents the probability of topic k in document n. In standard LDA, no prior distribution is assumed on $\mathbf{W}$, though it is common practice to assume Dirichlet distributions column-wise. The observation model is multinomial, with a total "budget" of L_n words to distribute in each document n. In contrast, the binary data models presented in this article are based on independent Bernoulli observations, and cannot

be recast as multinomial. An LDA-like model may arise if we assume that each column of $\mathbf{V}$ contains only one 1, and the rest are zeros (i.e. $L_n = 1$ for all n). This setting is however not considered in the article.

3.3 Bayesian inference

In this paper, we opt for Bayesian inference of the latent factors $\mathbf{W}$ and $\mathbf{H}$. This means that we seek to characterize the posterior distribution $p(\mathbf{W}, \mathbf{H}|\mathbf{V})$. The posterior in the considered models is not available in closed form and we will resort to numerical approximations (Markov Chain Monte Carlo sampling, variational inference). Another possible route is to seek point estimates through either constrained maximum likelihood (ML) or maximum a posteriori (MAP) estimation. For example, given Eq. (11) and the set of constraints (c1), maximum likelihood estimation writes

$$\min_{\mathbf{W},\mathbf{H}} -\log p(\mathbf{V}|\mathbf{WH}) = \sum_{f,n} v_{fn} \log([\mathbf{WH}]_{fn}) + (1 - v_{fn}) \log(1 - [\mathbf{WH}]_{fn})$$
$$\text{subject to} \quad (1)\ w_{fk}, h_{kn} \geq 0, \quad (2)\ \sum_k w_{fk} = 1, \quad (3)\ h_{kn} \in [0, 1]. \tag{14}$$

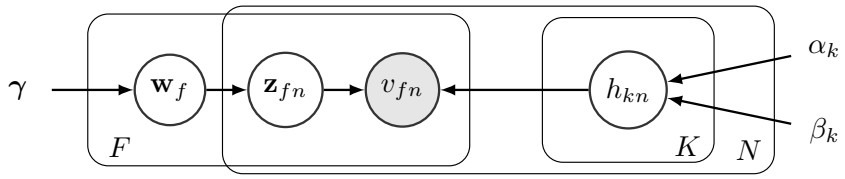
Such optimization problems may be tackled with, e.g., majorization-minimization (Bingham et al., 2009) or proximal gradient descent (Udell et al., 2016). Though it can be computationally more costly, we favored Bayesian inference for the following reasons. Characterizing the full posterior provides a measure of uncertainty over the latent parameters and over predicted values of $\mathbf{V}$. As we will show later, it allows in turn to infer the rank K of the factorization, a very desirable property that is more difficult to obtain with point estimation methods. Finally, Bayesian inference is rather customary in topic models such as LDA or Discrete Component Analysis (Buntine and Jakulin, 2006), and our work intends to follow similar principles.

4 Inference in the `Beta-Dir` model

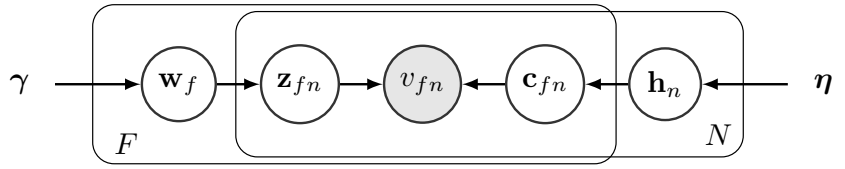
In this section, we derive a collapsed Gibbs sampler (Liu, 1994) for the `Beta-Dir` model. First, we will augment the model with latent indicator variables $\mathbf{Z}$ so that it becomes conjugate. Then the collapsed Gibbs sampler consists in marginalizing out the factors $\mathbf{W}$ and $\mathbf{H}$, thus running a Gibbs sampler over the indicator variables $\mathbf{Z}$ only. The interest of collapsed Gibbs sampling is that it offers improved mixing properties of the Markov chains, i.e., better exploration of the parameter space, thanks to the reduced dimensionality. We use a superscript, as in $x^{(j)}$, to indicate the j-th sample of a chain (after burn-in). After sampling, given a collection of samples $\mathbf{Z}^{(j)}$ from the posterior, we will be able to directly sample from the posteriors of interest $p(\mathbf{W}|\mathbf{V})$ and $p(\mathbf{H}|\mathbf{V})$.

4.1 Augmented model

We can augment the `Beta-Dir` model with indicator variables $\mathbf{z}_{fn}$, that contain component assignments from the Dirichlet factor, as shown in Bingham et al. (2009). More precisely, $\mathbf{z}_{fn}$ is a vector of dimension K with elements $z_{fkn} \in \{0, 1\}$ such that only one element equals to one and all the others equal to zero. In other words, $\mathbf{z}_{fn} \in \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$, where $\mathbf{e}_k$ is the k-th



(a) Partially augmented `Beta-Dir` model



(b) Fully augmented `Dir-Dir` model

Figure 3 Augmented models

canonical vector of $\mathbb{R}^K$. The augmented model is a mixture model described by:[3]

$$h_{kn} \sim \text{Beta}(\alpha_k, \beta_k) \tag{15}$$

$$\mathbf{w}_f \sim \text{Dirichlet}(\boldsymbol{\gamma}) \tag{16}$$

$$\mathbf{z}_{fn}|\mathbf{w}_f \sim \text{Discrete}(\mathbf{w}_f) \tag{17}$$

$$v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn} \sim \text{Bernoulli}\left(\prod_k h_{kn}^{z_{fkn}}\right). \tag{18}$$

Indeed, marginalizing $\mathbf{z}_{fn}$ from Eqs. (17)-(18) leads to Eq. (11) as shown next. From Eqs. (17)-(18) we have

$$p(v_{fn}|\mathbf{w}_f, \mathbf{h}_n) = \sum_k p(\mathbf{z}_{fn} = \mathbf{e}_k|\mathbf{w}_f)\,\text{Bernoulli}(v_{fn}|h_{kn}, z_{fkn}) \tag{19}$$

$$= \sum_k w_{fk}\, h_{kn}^{v_{fn}} (1 - h_{kn})^{1-v_{fn}}, \tag{20}$$

and thus $p(v_{fn} = 1|\mathbf{w}_f, \mathbf{h}_n) = [\mathbf{WH}]_{fn}$ and $p(v_{fn} = 0|\mathbf{w}_f, \mathbf{h}_n) = 1 - [\mathbf{WH}]_{fn}$, i.e., v_{fn} has the marginal distribution given by Eq. (11). A graphical representation of the augmented model is shown in Fig. 3-(a). Let us think of a recommender system application, where columns of $\mathbf{V}$ are users, and rows are items. An interpretation of the above model is the following. Each item f is characterized by a probability over topics, $\mathbf{w}_f$. Then, for each user-item pair, a topic k (indicated by $\mathbf{z}_{fn}$) is activated with probability w_{fk}, and the probability that the user n consumes this item ($v_{fn} = 1$) is h_{kn}.

Denoting by $\mathbf{Z}$ the $F \times K \times N$ tensor with entries z_{fkn}, the joint probability in the augmented model is given by:

$$p(\mathbf{V}, \mathbf{Z}, \mathbf{W}, \mathbf{H}) = p(\mathbf{W})p(\mathbf{Z}|\mathbf{W})p(\mathbf{H})p(\mathbf{V}|\mathbf{H}, \mathbf{Z}) =$$

$$\left[\prod_{f=1}^{F}\left(p(\mathbf{w}_f)\prod_{n=1}^{N} p(\mathbf{z}_{fn}|\mathbf{w}_f)\right)\right]\left[\prod_{n=1}^{N}\left(\prod_{k=1}^{K} p(h_{kn})\prod_{f=1}^{F} p(v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn})\right)\right]. \tag{21}$$

[3] Some readers may be more accustomed to the alternative notation where the "one-hot" variable $\mathbf{z}_{fn}$ is replaced by an integer-valued index $z_{fn} \in \{1, \ldots, K\}$. In this case, the Bernoulli parameter in Eq. (18) becomes $h_{z_{fn}n}$.

4.2 Collapsed Gibbs sampler

Thanks to the previous augmentation, and exploiting conjugacy, we can now marginalize out $\mathbf{W}$ and $\mathbf{H}$ from Eq. (21). The marginalized distribution has the following structure:

$$p(\mathbf{V}, \mathbf{Z}) = \prod_f \overbrace{\int p(\mathbf{w}_f) \prod_n p(\mathbf{z}_{fn}|\mathbf{w}_f)\, d\mathbf{w}_f}^{p(\underline{\mathbf{Z}}_f)} \prod_n \overbrace{\int \prod_k p(h_{kn}) \prod_f p(v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn})\, d\mathbf{h}_n}^{p(\mathbf{v}_n|\mathbf{Z}_n)}, \tag{22}$$

where $\underline{\mathbf{Z}}_f$ denotes the $K \times N$ matrix with entries $\{z_{fkn}\}_{kn}$, $\mathbf{Z}_n$ denotes the $F \times K$ matrix with entries $\{z_{fkn}\}_{fk}$. Let us define the following four variables that act as counters:

$$L_{fk} = \sum_n z_{fkn}, \qquad M_{kn} = \sum_f z_{fkn},$$
$$A_{kn} = \sum_f z_{fkn} v_{fn}, \qquad B_{kn} = \sum_f z_{fkn} \bar{v}_{fn}.$$

where $\bar{v}_{fn} = 1 - v_{fn}$. A_{kn} and B_{kn} count how many times the component k is associated to a "positive" observation ($v_{fn} = 1$) and to a "negative" observation ($\bar{v}_{fn} = 1$) in the n-th sample. Then we have the following expressions (derivations are available in Appendix B.1):

$$p(\underline{\mathbf{Z}}_f) = \frac{\Gamma(\sum_k \gamma_k)}{\prod_k \Gamma(\gamma_k)} \frac{\prod_k \Gamma(\gamma_k + L_{fk})}{\Gamma(\sum_k \gamma_k + N)} \tag{23}$$

$$p(\mathbf{v}_n|\mathbf{Z}_n) = \prod_k \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \frac{\Gamma(\alpha_k + A_{kn})\Gamma(\beta_k + B_{kn})}{\Gamma(\alpha_k + \beta_k + M_{kn})}. \tag{24}$$

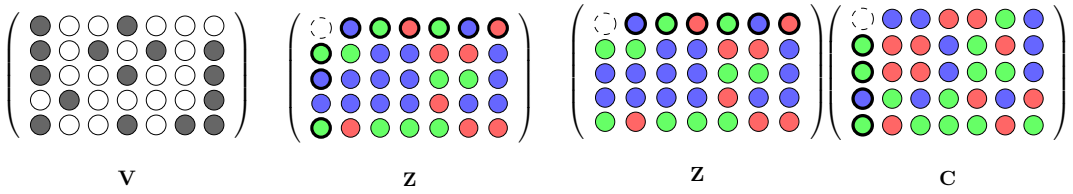
The posterior of the indicator variables $p(\mathbf{Z}|\mathbf{V})$ is not available in closed form and the proposed collapsed Gibbs sampler consists in iteratively sampling each vector $\mathbf{z}_{fn}$ given the current value of the other indicator vectors. Let $L_{fk}^{\neg fn}$, $M_{kn}^{\neg fn}$, $A_{kn}^{\neg fn}$, $B_{kn}^{\neg fn}$ be the state of the counters when the tube (f, n) of the tensor $\mathbf{Z}$ is left out of the sums:

$$L_{fk}^{\neg fn} = L_{fk} - z_{fkn}, \qquad M_{kn}^{\neg fn} = M_{kn} - z_{fkn},$$
$$A_{kn}^{\neg fn} = A_{kn} - z_{fkn} v_{fn}, \qquad B_{kn}^{\neg fn} = B_{kn} - z_{fkn} \bar{v}_{fn}.$$

In Appendix B.2 we show that the conditional posterior of $\mathbf{z}_{fn}$ given the remaining variables $\mathbf{Z}_{\neg fn}$ is given by:

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{V}) \propto \prod_k \left[(\gamma_k + L_{fk}^{\neg fn}) \frac{(\alpha_k + A_{kn}^{\neg fn})^{v_{fn}} (\beta_k + B_{kn}^{\neg fn})^{\bar{v}_{fn}}}{\alpha_k + \beta_k + M_{kn}^{\neg fn}} \right]^{z_{fkn}}. \tag{25}$$

This expression needs to be normalized to ensure a valid probability distribution. This can be easily done by computing the right-hand side of Eq. (25) for every of the K possible values of $\mathbf{z}_{fn}$ and normalizing by the sum. Eq. (25) shows that the probability of choosing a component k depends on the number of elements already assigned to that component. More precisely, it depends on the one hand on the number of elements assigned to component k in column n. On the other hand, it also depends on the proportion of elements in row f assigned to component k that explain ones (if $v_{fn} = 1$) or zeros (if $v_{fn} = 0$) in $\mathbf{V}$ in the total number of elements



(a) Observed matrix **(b)** `Beta-Dir` **(c)** `Dir-Dir`

Figure 4 Illustration of the Gibbs samplers. Colors represent component assignments (a value of k). When sampling an element (dashed circle) the probability of each component depends on the number of elements with thick circles in the same row or column that are currently assigned to that component.

Algorithm 1: Collapsed Gibbs sampler for `Beta-Dir`

Input: Observed matrix $\mathbf{V} \in \{0,1\}^{F \times N}$
Parameters: $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$
Output: Samples $\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(J)}$
Initialize: Random initialization of $\mathbf{Z}$
for $j = 1$ **to** J **do**
  for $f = 1$ **to** F **do**
    for $n = 1$ **to** N **do**
      if v_{fn} *not missing* **then**
        Sample $\mathbf{z}_{fn}^{(j)} \sim p(\mathbf{z}_{fn} | \mathbf{Z}_{\neg fn}, \mathbf{V})$ (Eq.(25))
      end
    end
  end
end

associated to k in that row (see Figure 4). The parameters $\gamma_k, \alpha_k, \beta_k$ act as pseudo-counts: they give *a priori* belief about how many elements are assigned to each component.

Our collapsed Gibbs sampling is summarized in Alg. (1). Note that, although Alg. (1) does not explicitly include it, we must draw samples during an initial burn-in phase (as required by any MCMC method) before collecting the last J samples, after the chain has converged to the stationary distribution. Note also that the algorithm can readily deal with incomplete matrices by simply skipping missing entries (i.e., the loop over f and n only runs over available entries).

Latent factors posteriors. Thanks to conjugacy, the conditional posteriors of $\mathbf{W}$ and $\mathbf{H}$ given $\mathbf{Z}$ and $\mathbf{V}$ are given by:

$$\mathbf{w}_f | \underline{\mathbf{Z}}_f \sim \text{Dirichlet}(\boldsymbol{\gamma} + \sum_n \mathbf{z}_{fn}) \tag{26}$$

$$h_{kn} | \mathbf{Z}_n, \mathbf{v}_n \sim \text{Beta}(\alpha_k + A_{kn},\ \beta_k + B_{kn}). \tag{27}$$

The conditional posterior expectations are given by:

$$\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\underline{\mathbf{Z}}_f] = \frac{\boldsymbol{\gamma} + \sum_n \mathbf{z}_{fn}}{\sum_k \gamma_k + \sum_k \sum_n z_{fkn}} = \frac{\boldsymbol{\gamma} + \sum_n \mathbf{z}_{fn}}{\sum_k \gamma_k + N} \tag{28}$$

$$\mathbb{E}_{h_{kn}}[h_{kn}|\mathbf{Z}_n, \mathbf{v}_n] = \frac{\alpha_k + A_{kn}}{\alpha_k + A_{kn} + \beta_k + B_{kn}} = \frac{\alpha_k + \sum_f z_{fkn} v_{fn}}{\alpha_k + \beta_k + \sum_f z_{fkn}}, \tag{29}$$

where we used the equalities $\sum_{kn} z_{fkn} = N$ and $A_{kn} + B_{kn} = \sum_f z_{fkn}$. Using the law of total expectation, i.e., $\mathbb{E}[X] = \mathbb{E}_Y[\mathbb{E}_X[X|Y]]$, and given a set a samples $\mathbf{Z}^{(j)}$, it follows that the marginal posterior expectations of the latent factors can be computed as:

$$\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\mathbf{V}] = \mathbb{E}_{\mathbf{Z}}[\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\mathbf{V}, \mathbf{Z}]] = \mathbb{E}_{\underline{\mathbf{Z}}_f}[\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\underline{\mathbf{Z}}_f]] \approx \frac{\boldsymbol{\gamma} + \frac{1}{J}\sum_j \sum_n \mathbf{z}_{fn}^{(j)}}{\sum_k \gamma_k + N} \tag{30}$$

$$\mathbb{E}_{h_{kn}}[h_{kn}|\mathbf{V}] = \mathbb{E}_{\mathbf{Z}_n}[\mathbb{E}_{h_{kn}}[h_{kn}|\mathbf{Z}_n, \mathbf{v}_n]] \approx \frac{1}{J}\sum_j \frac{\alpha_k + \sum_f z_{fkn}^{(j)} v_{fn}}{\alpha_k + \beta_k + \sum_f z_{fkn}^{(j)}}. \tag{31}$$

Prediction. The predictive posterior distribution of an unseen data sample v_{fn}^* given the available data $\mathbf{V}$ is given by

$$\begin{aligned} p(v_{fn}^*|\mathbf{V}) &= \int p(v_{fn}^*|\mathbf{w}_f, \mathbf{h}_n) p(\mathbf{w}_f, \mathbf{h}_n|\mathbf{V})\, d\mathbf{w}_f d\mathbf{h}_n \\ &= \mathbb{E}[\mathbf{w}_f \mathbf{h}_n|\mathbf{V}]^{v_{fn}^*} (1 - \mathbb{E}[\mathbf{w}_f \mathbf{h}_n|\mathbf{V}])^{1 - v_{fn}^*}. \end{aligned} \tag{32}$$

Because the predictive posterior is a Bernoulli distribution, its expectation is given by $\mathbb{E}[\mathbf{w}_f \mathbf{h}_n|\mathbf{V}]$, which can be approximated using samples $\mathbf{W}^{(j)}$, $\mathbf{H}^{(j)}$ from the distributions given by Eqs. (26)-(27) given $\mathbf{Z} = \mathbf{Z}^{(j)}$:

$$\mathbb{E}[v_{fn}^*|\mathbf{V}] = \mathbb{E}[\mathbf{w}_f \mathbf{h}_n|\mathbf{V}] \approx \frac{1}{J}\sum_j \mathbf{w}_f^{(j)} \mathbf{h}_n^{(j)}. \tag{33}$$

4.3 Collapsed variational inference

Given the collapsed model of Eqs. (22)-(24) we may derive a mean-field Collapsed Variational Bayes algorithm (CVB) (Teh et al., 2007) by assuming that the posterior factorizes as $q(\mathbf{Z}) = \prod_{fn} q(\mathbf{z}_{fn})$. The key of CVB is that its free energy is a strictly better bound on the evidence than the free energy of the standard, i.e., uncollapsed, VB. We compute the CVB updates by applying the mean-field VB updates to the collapsed model:

$$q(\mathbf{z}_{fn}|\mathbf{V}) \propto \exp\{\mathbb{E}_{q(\mathbf{Z}_{\neg fn})}[\log p(\mathbf{V}, \mathbf{Z})]\}, \tag{34}$$

where the expectations are taken over the variational posterior. This leads us to

$$\begin{aligned} &q(\mathbf{z}_{fn}|\mathbf{V}) \\ &\propto \prod_k \exp\left\{ \mathbb{E}_q[\log(\gamma_k + L_{fk}^{\neg fn})] \frac{\mathbb{E}_q[\log(\alpha_k + A_{kn}^{\neg fn})]^{v_{fn}} \mathbb{E}_q[\log(\beta_k + B_{kn}^{\neg fn})]^{\bar{v}_{fn}}}{\mathbb{E}_q[\log M_{kn}^{\neg fn}]} \right\}. \end{aligned} \tag{35}$$

The expectations of the form $\mathbb{E}_{q(z)}[\log(x+z)]$ are expensive to compute. A simpler alternative is CVB0 (Asuncion et al., 2009), which uses a zero-order Taylor approximation $\mathbb{E}_q(z)[\log(x+z)] \approx$

$\log(x + \mathbb{E}_{q(z)}[z])]$ and has been shown to give, in some cases, better inference results than CVB. Under the CVB0 version our update becomes

$$q(\mathbf{z}_{fn}|\mathbf{V}) \propto \prod_k (\gamma_k + \mathbb{E}_q[L_{fk}^{\neg fn}]) \frac{(\alpha_k + \mathbb{E}_q[A_{kn}^{\neg fn}])^{v_{fn}} (\beta_k + \mathbb{E}_q[B_{kn}^{\neg fn}])^{\bar{v}_{fn}}}{\alpha_k + \beta_k + \mathbb{E}_q[M_{kn}^{\neg fn}]}, \tag{36}$$

which has a similar structure to the collapsed Gibbs sampler in Eq. (25). Overall, the collapsed VB algorithm has the same structure as the Gibbs sampler summarized in Alg. (1). Note that when the data matrix is too large for batch processing, one can routinely resort to stochastic variational inference (Hoffman et al., 2013).

Latent factors posteriors. The variational distributions of the factors can be obtained from the uncollapsed version:

$$q(\mathbf{w}_f) = \text{Dirichlet}(\boldsymbol{\gamma} + \sum_n \mathbb{E}_q[\mathbf{z}_{fn}]) \tag{37}$$

$$q(h_{kn}) = \text{Beta}(\alpha_k + \mathbb{E}_q[A_{kn}], \beta_k + \mathbb{E}_q[B_{kn}])). \tag{38}$$

The Taylor approximation breaks the theoretical guarantees of their superiority over the ones given by uncollapsed VB. Still, they have been reported to work better than VB in practice. Another drawback of the approximation is the loss of convergence guarantees. Although we do not address this issue here, this has been recently addressed by Ishiguro et al. (2017), where an annealing strategy is used to gradually decrease the portion of the variational posterior changes.

Prediction. The predictive posterior can be computed as in Eq. (32), and its expectation is computed using the variational approximations of the factors, i.e.,

$$\mathbb{E}[v_{fn}^*|\mathbf{V}] = \mathbb{E}[\mathbf{w}_f \mathbf{h}_n|\mathbf{V}] \approx \mathbb{E}_{q(\mathbf{w}_f)}[\mathbf{w}_f]\, \mathbb{E}_{q(\mathbf{h}_n)}[\mathbf{h}_n]. \tag{39}$$

4.4 Approximating infinite components

Recall that in the augmented model, the component assignments $\mathbf{z}_{fn}$ have a Discrete distribution such that (Eqs. (16)-(17))

$$\mathbf{w}_f \sim \text{Dirichlet}(\boldsymbol{\gamma})$$
$$\mathbf{z}_{fn}|\mathbf{w}_f \sim \text{Discrete}(\mathbf{w}_f).$$

The variable $\mathbf{w}_f$ may be integrated out leading to the expression of $p(\underline{\mathbf{Z}}_f)$ given by Eq.(23). In Appendix B.2, we show that the prior conditionals are given by:

$$p(\mathbf{z}_{fn} = \mathbf{e}_k|\mathbf{Z}_{\neg fn}) = \frac{\gamma_k + L_{fk}^{\neg fn}}{\sum_k \gamma_k + N - 1}. \tag{40}$$

Let us assume from now that the Dirichlet prior parameters are such that $\gamma_k = \gamma/K$, where γ is a fixed nonnegative scalar, so that:

$$p(\mathbf{z}_{fn} = \mathbf{e}_k|\mathbf{Z}_{\neg fn}) = \frac{\gamma/K + L_{fk}^{\neg fn}}{\gamma + N - 1}. \tag{41}$$

The conditional prior given by Eq. (41) is reminiscent of the Chinese Restaurant process (CRP) (Aldous, 1985; Anderson, 1991; Pitman, 2002). In the limit when $K \to \infty$, the probability of

assigning $\mathbf{z}_{fn}$ to component k is proportional to the number $L_{fk}^{\neg fn}$ of current assignments to that component. Let K^+ denote the current number of non-empty components (i.e., such that $L_{fk}^{\neg fn} > 0$). Then the probability of choosing an empty component is

$$p(\mathbf{z}_{fn} = \mathbf{e}_k | \mathbf{Z}_{\neg fn}, L_{fk}^{\neg fn} = 0) = \lim_{K \to \infty} (K - K^+) \frac{\gamma/K}{\gamma + N - 1} = \frac{\gamma}{\gamma + N - 1}. \tag{42}$$

Note that the latter probability does not depend on K. In practice, we set K to a large value and observed self-pruning of the number of components, hence achieving to automatic order selection, similar to Hoffman et al. (2010). Implementing exact inference in the truly nonparametric model

$$\underline{\mathbf{Z}}_f \sim \text{CRP}(\gamma) \tag{43}$$

$$\mathbf{v}_n | \mathbf{Z}_n \sim p(\mathbf{v}_n | \mathbf{Z}_n) \tag{44}$$

is more challenging. This is because there is a CRP for each feature f, and some empty components may become unidentifiable in the limit. This is a known issue that could be addressed using for example a Chinese Restaurant Franchise process (Teh et al., 2006) but is beyond the scope of this article.

5 Inference in the Dirichlet-Dirichlet model

The methodology to obtain a collapsed Gibbs sampler for the `Dir-Dir` model is similar to the approach followed for the `Beta-Dir` model. It is possible to augment the model with the same auxiliary variable $\mathbf{z}_{fn}$ and compute the expression of $p(\mathbf{z}_{fn} | \mathbf{Z}_{\neg fn}, \mathbf{V})$ in closed form. However, the expression of the conditional posterior, given in Appendix C, involves combinatorial computations and is infeasible in practice. As such, we propose an alternative Gibbs sampler that relies on a double augmentation, presented next. Obtaining a variational collapsed algorithm for the `Dir-Dir` model is not straightforward, even using the double augmentation, and is left for future work.

5.1 Fully augmented model

Unlike the `Beta-Dir` model, the `Dir-Dir` model is not fully conjugate after a first augmentation. We propose a second augmentation with a new indicator variable $\mathbf{c}_{fn} \in \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$, that plays a similar role to $\mathbf{z}_{fn}$ The *fully* augmented version is:

$$\mathbf{h}_n \sim \text{Dirichlet}(\boldsymbol{\eta}) \tag{45}$$

$$\mathbf{w}_f \sim \text{Dirichlet}(\boldsymbol{\gamma}) \tag{46}$$

$$\mathbf{c}_{fn} | \mathbf{h}_n \sim \text{Discrete}(\mathbf{h}_n) \tag{47}$$

$$\mathbf{z}_{fn} | \mathbf{w}_f \sim \text{Discrete}(\mathbf{w}_f) \tag{48}$$

$$v_{fn} = \sum_k c_{fkn} z_{fkn}. \tag{49}$$

To show that this is a valid augmentation, note that v_{fn} can only be nonzero (and equal to 1) if $\mathbf{c}_{fn} = \mathbf{z}_{fn}$. Then, the marginal probability of $v_{fn} = 1$ is given by

$$p(v_{fn} = 1|\mathbf{w}_f, \mathbf{h}_n) = \sum_k p(v_{fn} = 1, \mathbf{z}_{fn} = \mathbf{c}_{fn} = \mathbf{e}_k|\mathbf{w}_f, \mathbf{h}_n) \tag{50}$$

$$= \sum_k p(\mathbf{z}_{fn} = \mathbf{e}_k|\mathbf{w}_f)p(\mathbf{c}_{fn} = \mathbf{e}_k|\mathbf{h}_n) \tag{51}$$

$$= \sum_k w_{fk}h_{kn}, \tag{52}$$

and we thus recover the Bernoulli model of Eq. (11) as announced. Compared to the `Beta-Dir` model and using our recommender system analogy, this means that, in each user-item pair, the user also activates one topic, and then consumes the item if the user active topic is equal to the item active topic. The `Dir-Dir` model makes a stronger assumption than the `Beta-Dir` since the user can only activate one topic per item. A graphical representation of the fully augmented model is given in Fig. 3-(b). In the following, we denote by $\mathbf{C}$ the $F \times K \times N$ tensor with entries c_{fkn}, and by $\mathbf{C}_n$ the $F \times K$ matrix with entries $\{c_{fkn}\}_{fk}$.

5.2 Collapsed Gibbs sampling

In this section we show that $\mathbf{W}$ and $\mathbf{H}$ can be marginalized from the joint probability of the fully augmented model and then propose a collapsed Gibbs sampler for $p(\mathbf{Z}, \mathbf{C}|\mathbf{V})$. The joint probability is given by:

$$p(\mathbf{V}, \mathbf{Z}, \mathbf{C}) = \prod_{f,n} p(v_{fn}|\mathbf{z}_{fn}, \mathbf{c}_{fn}) \prod_f p(\underline{\mathbf{Z}}_f) \prod_n p(\mathbf{C}_n), \tag{53}$$

where

$$p(\underline{\mathbf{Z}}_f) = \frac{\Gamma(\sum_k \gamma_k)}{\prod_k \Gamma(\gamma_k)} \frac{\prod_k \Gamma(\gamma_k + \sum_n z_{fkn})}{\Gamma(\sum_k \gamma_k + N)} \tag{54}$$

$$p(\mathbf{C}_n) = \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \frac{\prod_k \Gamma(\eta_k + \sum_f c_{fkn})}{\Gamma(\sum_k \eta_k + F)} \tag{55}$$

$$p(v_{fn}|\mathbf{z}_{fn}, \mathbf{c}_{fn}) = \delta(v_{fn} - \sum_k c_{fkn}z_{fkn}), \tag{56}$$

and where δ denotes the Dirac delta function. Following Section 4.2 and Appendix B.2, the prior conditional are given by:

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}) \propto \prod_k (\gamma_k + L_{fk}^{\neg fn})^{z_{fkn}} \tag{57}$$

$$p(\mathbf{c}_{fn}|\mathbf{C}_{\neg fn}) \propto \prod_k (\eta_k + Q_{kn}^{\neg fn})^{c_{fkn}}. \tag{58}$$

where $Q_{kn}^{\neg fn} = \sum_{f' \neq f} c_{f'kn}$ and $L_{fk}^{\neg fn} = \sum_{n' \neq n} z_{fkn'}$ is as before. When $v_{fn} = 1$, $\mathbf{c}_{fn}$ and $\mathbf{z}_{fn}$ must be assigned to the same component ($\mathbf{c}_{fn} = \mathbf{z}_{fn}$). To respect this constraint, we may sample them together from the posterior. Introducing the vector $\mathbf{x}_{fn}$ such that $\mathbf{x}_{fn} = \mathbf{z}_{fn} = \mathbf{c}_{fn}$, the conditional posterior is given by:

$$p(\mathbf{x}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{C}_{\neg fn}, v_{fn} = 1) \propto \prod_k \left[(\gamma_k + L_{fk}^{\neg fn})(\eta_k + Q_{kn}^{\neg fn})\right]^{x_{fkn}}. \tag{59}$$

Algorithm 2: Collapsed Gibbs sampler for `Dir-Dir`

Input: Observed matrix $\mathbf{V} \in \{0,1\}^{F \times N}$
Parameters: $\boldsymbol{\gamma}, \boldsymbol{\eta}$
Output: Samples $\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(J)}$, $\mathbf{C}^{(1)}, ..., \mathbf{C}^{(J)}$
Initialize: Random initialization of $\mathbf{Z}$ and $\mathbf{C}$
for $j = 1$ **to** J **do**
  for $f = 1$ **to** F **do**
    for $n = 1$ **to** N **do**
      if v_{fn} *not missing* **then**
        if $v_{fn} = 1$ **then**
          Sample $\mathbf{x} \sim p(\mathbf{x}|\mathbf{Z}_{\neg fn}, \mathbf{C}_{\neg fn}, \mathbf{V})$ (Eq. (59))
          $\mathbf{z}_{fn}^{(j)} = \mathbf{x}$
          $\mathbf{c}_{fn}^{(j)} = \mathbf{x}$
        else
          Sample $\mathbf{z}_{fn}^{(j)} \sim p(z_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{C}, \mathbf{V})$ (Eq. (60))
          Sample $\mathbf{c}_{fn}^{(j)} \sim p(z_{fn}|\mathbf{Z}, \mathbf{C}_{\neg fn}, \mathbf{V})$ (Eq. (61))
        end
      end
    end
  end
end

When $v_{fn} = 0$, we can assign to one of the two auxiliary variables any component not currently assigned to the other auxiliary variable. The respective conditional posteriors are given by:

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{c}_{fn}, v_{fn} = 0) \propto \prod_k \left[(\gamma_k + L_{fk}^{\neg fn})(1 - c_{fkn})\right]^{z_{fkn}} \tag{60}$$

$$p(\mathbf{c}_{fn}|\mathbf{z}_{fn}, \mathbf{C}_{\neg fn}, v_{fn} = 0) \propto \prod_k \left[(\eta_k + Q_{kn}^{\neg fn})(1 - z_{fkn})\right]^{c_{fkn}}. \tag{61}$$

A pseudo-code of the resulting Gibbs sampler is given in Alg. (2). As with the `Beta-Dir` model, we set $\gamma_k = \gamma/K$ with K large to emulate a nonparametric setting (note that $\boldsymbol{\eta}$ does not need to depend on K itself).

Latent factors posteriors and prediction. The conditional posteriors of the latent factors given $\mathbf{Z}$ and $\mathbf{C}$ are given by:

$$\mathbf{w}_f|\underline{\mathbf{Z}}_f \sim \text{Dirichlet}(\boldsymbol{\gamma} + \sum_n \mathbf{z}_{fn}) \tag{62}$$

$$\mathbf{h}_n|\mathbf{C}_n \sim \text{Dirichlet}(\boldsymbol{\eta} + \sum_f \mathbf{c}_{fn}). \tag{63}$$

As done with the `Beta-Dir` model, we may use the law of total expectation and the samples $\mathbf{z}_{fn}^{(j)}$

to obtain Monte-Carlo estimates of the posterior expectations:

$$\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\mathbf{V}] = \mathbb{E}_{\mathbf{Z}}[\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\mathbf{V},\mathbf{Z}]] = \mathbb{E}_{\underline{\mathbf{Z}}_f}[\mathbb{E}_{\mathbf{w}_f}[\mathbf{w}_f|\underline{\mathbf{Z}}_f]] \approx \frac{\boldsymbol{\gamma} + \frac{1}{J}\sum_j \sum_n \mathbf{z}_{fn}^{(j)}}{\sum_k \gamma_k + N} \tag{64}$$

$$\mathbb{E}_{\mathbf{h}_n}[\mathbf{h}_n|\mathbf{V}] = \mathbb{E}_{\mathbf{C}}[\mathbb{E}_{\mathbf{h}_n}[\mathbf{h}_n|\mathbf{V},\mathbf{C}]] = \mathbb{E}_{\mathbf{C}_n}[\mathbb{E}_{\mathbf{h}_n}[\mathbf{h}_n|\mathbf{C}_n]] \approx \frac{\boldsymbol{\eta} + \frac{1}{J}\sum_j \sum_f \mathbf{c}_{fn}^{(j)}}{\sum_k \eta_k + F}. \tag{65}$$

As in the `Beta-Dir` model, $\mathbf{W}$ and $\mathbf{H}$ can be sampled in a second step given a collection of samples of $\mathbf{Z}$ and $\mathbf{C}$. The predictive posterior and its expectation can be computed as in Eqs. (32), (33).

6 Experiments

We show the performance of the proposed NBMF methods for different tasks in multiple datasets. The datasets, the algorithms, and scripts to replicate all the reported results are available through our R package.

6.1 Datasets

We consider five different public datasets, described next and displayed in Fig. 5.

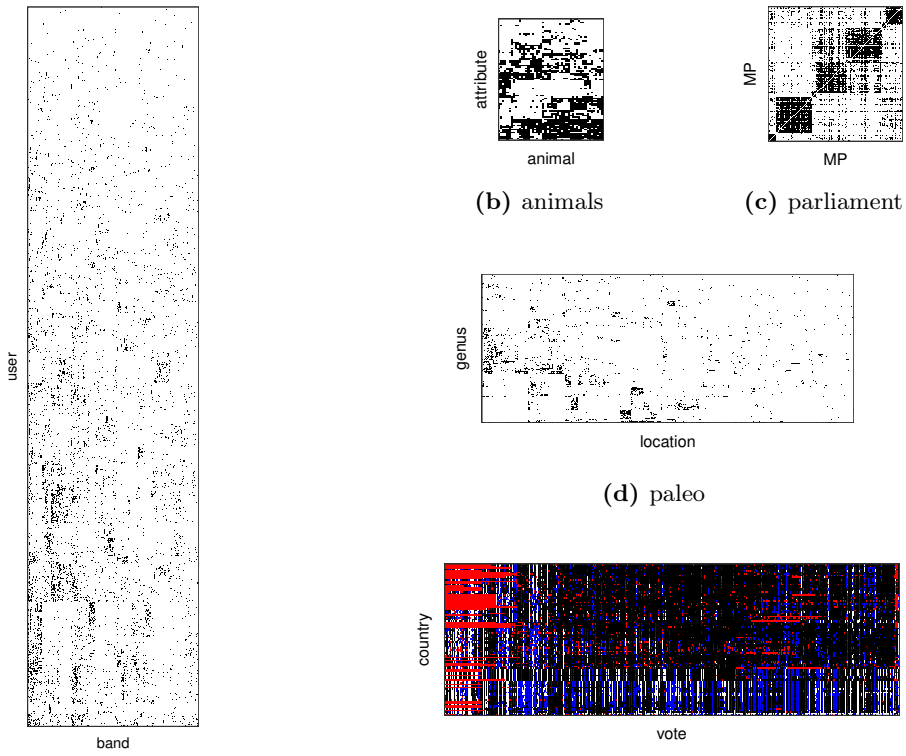
Animals (`animals`). The animals dataset (Kemp et al., 2006) contains 50 animals and 85 binary attributes such as *nocturnal*, *hibernates*, *small* or *fast*. The matrix takes $v_{fn} = 1$ if animal n has attribute f.

Last.fm (`lastfm`). We use a binarized subset of the Last.fm dataset (Celma, 2010) where rows correspond to users and columns correspond to musical artists. The matrix takes $v_{fn} = 1$ if user n has listened to artist f at least once. The matrix has $F = 285$ rows and $N = 1226$ columns.

Paleontological data (`paleo`). The NOW (New and Old Worlds) fossil mammal database contains information of fossils found in specific paleontological sites (NOW, 2018). From the original paleontological data, we build a matrix where each row is a genus, each column is a location, and $v_{fn} = 1$ if genus f has been found at location n. We used the same pre-processing as in Bingham et al. (2009) (i.e., we discarded small and infrequent genus, locations with only one genus and kept locations with longitude between 0 and 60 degrees East) and obtained a matrix with $F = 253$ rows and $N = 902$ columns.

Catalan parliament (`parliament`). We created a list of the current members of the Catalan parliament and collected the information of who follows whom on Twitter (March 2018). With this data, we created a square adjacency matrix where $v_{fn} = 1$ if member f follows member n. There are seven political groups represented. The resulting matrix has 135 rows and columns ($F = N$).

UN votes (`unvotes`). The United Nations General Assembly Voting Data is a dataset that contains the roll-call votes in the UN General Assembly between 1946-2017 (Voeten, 2013). Votes can be *yes*, *no*, or *abstention*. From this data, we created a matrix where $v_{fn} = 1$ if country f voted *yes* to the call n, $v_{fn} = 0$ if it voted *no*, and a missing value if the country did not participate in that call or was not a member of the UN at that time. Next, abstention votes



(a) lastfm **(b)** animals **(c)** parliament **(d)** paleo **(e)** unvote

Figure 5 Datasets. Black entries correspond to $v_{fn} = 1$ and white entries correspond to zero values. In the `unvote` dataset, blue entries represent "abstention" values and red entries are missing votes. For best visualization, rows and columns are re-ordered with complete linkage clustering, except for 1) `parliament` in which parliament members are sorted by parliamentary group and 2) `unvotes` where votes are sorted chronologically.

will be treated as either negative votes (*no*) or missing data, as specified in each experiment. The resulting matrix has $F = 200$ rows and $N = 5429$ columns.

6.2 Methods and setting

State-of-the-art methods. We compare our proposed methods with the following state-of-the-art methods for binary data.

`logPCA-K`. Probabilistic PCA with Bernoulli likelihood. We use the algorithm presented in Collins et al. (2002). The notation `logPCA-K` will embed the chosen number of components K (e.g., `logPCA-8` signifies $K = 8$). We used the R package `logisticPCA` (Landgraf and Lee, 2015) with default parameters.

`bICA-K`. The binary ICA method introduced in Kabán and Bingham (2008), which uses uncollapsed mean-field variational inference over the partially augmented model (Eqs. (15)-(18)). This is also a parametric method that requires setting K.

Proposed methods. Our proposed methods are as follows.

`Beta-Dir GS`. Estimation in the `Beta-Dir` model with collapsed Gibbs sampling. Beta pa-

rameters are set to $\alpha_k = \beta_k = 1$. To emulate a nonparametric setting, the Dirichlet parameters are set to $\gamma_k = 1/K$ and the number of components is set to $K = 100$.

`Beta-Dir VB`. Estimation in the `Beta-Dir` model with collapsed variational Bayes (CVB0). Beta parameters are set to $\alpha_k = \beta_k = 1$. To emulate a nonparametric setting, we set $\gamma_k = 1/K$ and $K = 100$.

`Dir-Dir GS`. Estimation in the `Dir-Dir` model with collapsed Gibbs sampling. To emulate a nonparametric setting, we set $\gamma_k = 1/K$, $\eta_k = 1$, and $K = 100$.

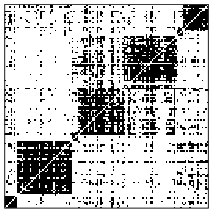
`c-bICA-K`. Collapsed bICA. The algorithm corresponds to `Beta-Dir VB` with $\alpha_k = \beta_k = \gamma_k = 1$. It is the collapsed version of `bICA-K` using CVB0 and *without* the nonparametric approximation.

Implementation details. For each dataset, we ran some preliminary experiments to assess the number of iterations needed by the algorithms to converge. For the Gibbs samplers, we set a conservative burn-in phase of 4,000 iterations and kept the last 1,000 samples of $\mathbf{Z}$ after burn-in. A total number of 500 iterations where used for the variational algorithms. In every experiment, we initialized the Gibbs samplers with a random tensor $\mathbf{Z}$ such that $\mathbf{z}_{fn} = \mathbf{e}_k$ with random k. We did not find a special sensitivity to the initial state of the Gibbs Sampler, but we chose a random initialization as a good practice. We also tried initializing with some variational steps, which is another common practice, but did not see significant improvements. Similarly, we initialized the variational algorithms with a random $\mathbb{E}[\mathbf{Z}]$ such that $\mathbb{E}[\mathbf{z}_{fn}] = \mathbf{e}_k$ with random k. Our variational algorithms are sensitive to the initial state when some components are empty. Because an empty component has a lower probability of being chosen, in practice the variational algorithms are not capable of refilling it again. Initializing from a random state with lots of used components is, therefore, a safer way to avoid these local maxima.

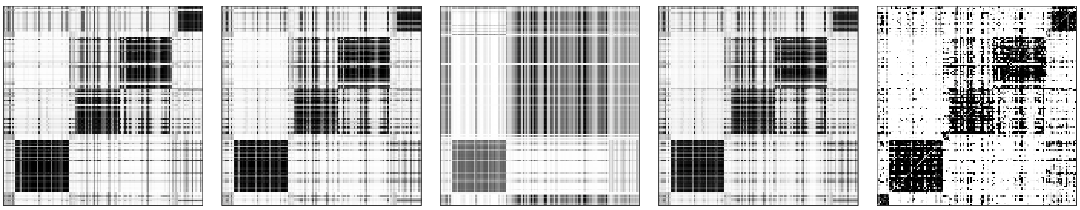
Estimators. The algorithms `Beta-Dir GS` and `Dir-Dir GS` return samples from the posterior of $p(\mathbf{W}, \mathbf{H}|\mathbf{V})$. Point estimates of the dictionary $\mathbf{W}$ and data expectation $\hat{\mathbf{V}} = \mathbf{WH}$ are computed by averaging (posterior mean) and by Eq. (33), respectively. `Beta-Dir VB`, `bICA` and `c-bICA` return variational approximations of the posterior of $\mathbf{W}$ and $\mathbf{H}$. Point estimates of $\mathbf{W}$ and $\hat{\mathbf{V}}$ are computed from the variational distribution mean and by Eq. (39). `logPCA` returns maximum likelihood (ML) estimates $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$. The data expectation is computed as $\hat{\mathbf{V}} = \sigma(\hat{\mathbf{W}}\hat{\mathbf{H}})$.

Sensitivity to hyperparameters. When using Dirichlet priors, inferences may be quite sensitive especially for small values of its concentration parameter (Steck and Jaakkola, 2003). For the setting described in Section 6.4, we have tested our algorithms under toy data generated from the model, setting the concentration parameter to 1. We have repeated different inferences with the concentration parameter of the estimator ranging from 0.1 to 10 and observed very small variations on the perplexity (around 0.1, and decreasing as the size of the observed data increases). Thus, for the sake of simplicity, we have therefore decided to set the concentrations parameters to 1 which corresponds to a uniform Dirichlet prior.

Computational cost. The time complexity of the collapsed algorithms is $\mathcal{O}(FKN)$ (assuming, for the sake of simplicity, that the Multinomial random number generator is $\mathcal{O}(1)$). Note that, unlike some non-mean-parameterized Bernoulli models (Zhou, 2015), or all Poisson models, zeros cannot be ignored because they represent another category, not a lack of observation or a zero-count.



(a) Original



(b) Beta-Dir GS **(c)** Beta-Dir VB **(d)** Dir-Dir GS **(e)** bICA-8 **(f)** logPCA-8

Figure 6 Reconstructed matrices for the `parliament` dataset.

6.3 Dictionary learning and data approximation

6.3.1 Experiments with the `parliament` dataset

First, we want to form an idea of how well the different models can fit original data. We focus on the `parliament` dataset, which has reasonable size and a clear structure. We applied the three proposed nonparametric methods `Beta-Dir GS`, `Beta-Dir VB`, `Dir-Dir GS` and the state-of-the-art methods `bICA` and `logPCA`. For each method, we compute the negative log-likelihood of the data approximation $\hat{\mathbf{V}}$, which serves as a measure of fit:

$$D(\mathbf{V}|\hat{\mathbf{V}}) = -\sum_{fn} \log p(v_{fn}|\hat{v}_{fn}). \tag{66}$$

`bICA` was run with increasing values of K and the fit ceased increasing for $K = 8$ which is the value used in the results (note that in this case $\hat{\mathbf{V}}$ is the posterior mean estimate and not the ML estimate, so the likelihood is not meant to increase monotonically). `logPCA` was run with the same value $K = 8$. The data approximations $\hat{\mathbf{V}}$ and dictionaries obtained with the different methods are displayed in Figs. 6 and 7, respectively.

In terms of data approximation, `Beta-Dir VB` achieves the best fit among the mean-parameterized models in terms of negative log-likelihood (4,729) followed by `Beta-Dir GS` (4,863). The dictionaries returned by these two algorithms are very similar, with only nine active components. `bICA-8` comes next in terms of fit (4,957). We also applied `bICA` with $K > 8$ components but this did not substantially improve the likelihood. `Dir-Dir GS` returns the worst fit (8,930), with only two active components. Overall `logPCA-8` returns the smallest negative log-likelihood (1,783). This due to its larger flexibility as compared to the mean-parameterized models (real-valued factors $\mathbf{W}$ and $\mathbf{H}$, with product $\mathbf{WH}$ mapped to $[0, 1]$). However, this is at the cost of meaningfulness of the decomposition, as shown in Fig. 7 and explained next.

In dictionary learning, we want to learn a meaningful decomposition of the data. The columns of the dictionary $\mathbf{W}$ are expected to contain *patterns* or *prototypes* characteristic of

the data. In particular, NMF is known to produce so-called part-based representations (each sample, a column of $\mathbf{V}$, is approximated as a constructive linear combination of building units) (Lee and Seung, 1999). When the rows of the dictionary are given Dirichlet priors, $\mathbf{w}_f$ can also be interpreted as the probability distribution of feature f over the K components. In Fig. 7, the rows of the dictionaries displayed correspond to members of the parliament (MP). For each MP we show its Twitter username and its political party. The dictionaries returned by the mean-parameterized factorization methods are easily interpretable. In particular, the dictionaries returned by `Beta-Dir GS`, `Beta-Dir VB` and to some extent `bICA-8` closely reflect the party memberships of the MPs. `Dir-Dir GS`, which is based on a less flexible model, only captures two sets of MPs, one with the members of *Cs* (the main opposition party) and the other with members of the remaining parties, regardless of political alignment (left-wing, right-wing, independentist and anti-independentist). In contrast, the dictionary returned by `logPCA-8` is much more difficult to interpret.

6.3.2 Experiments with the `unvotes` dataset

In this section, we consider a subset of `unvotes`, reduced to the 1946-1990 range which corresponds to the Cold War period. Furthermore, the abstentions are here treated as missing values. Fig. 8 shows the dictionaries learned by the five considered methods. As before, `bICA` was applied with various values of K and we selected the value that leads to the smallest negative log-likelihood ($K = 7$). Accordingly, `logPCA` was also applied with $K = 7$.

Fig. 8 shows that `Beta-Dir GS` returns the finest dictionary, detecting political blocks that tended to vote similarly in the UN assembly and capturing some nuances that the other algorithms do not find. European countries (and members or allies of NATO such as the USA, Japan, or Australia) are concentrated in one component, denoting similar voting strategies. The former members of the Soviet Union and the Warsaw Pact also form a block of their own, with some allies such as Cuba or the former Yugoslavia. Members of the Non-Aligned Movement (even countries that became members after 1991, such as Guatemala, Thailand, or Haiti), from Egypt to Cuba and from Honduras to Haiti, are split into two blocks: the Latin American group and the Asian-African group. Another detected alliance is between the United States and Israel, which are distributed between the European component and a component of their own. `Beta-Dir VB` detects the split between the Warsaw and NATO blocks, and the alliance between the USA and Israel, but it fails to detect the two subgroups of the Non-Aligned Movement, which is considered a single block. `bICA-7` returns similar results to `Beta-Dir GS` but fails to detect the alliance between the USA and Israel. Note that the results of `bICA` are obtained with a well-chosen value of K while `Beta-Dir GS` automatically detects a suitable value. The underlying assumption of `Dir-Dir` (one topic per country and one topic per vote) seems too simplistic for this dataset, and the algorithm puts every country in the same component. Again and as somewhat expected, the dictionary learned by `logPCA` is more difficult to interpret.

6.3.3 Experiments with the `paleo` dataset

We finally look into the dictionaries returned by the five considered method on the `paleo` dataset, see Fig. 9. The same strategy was applied to find a suitable value of K for `bICA` and `logPCA`, leading to $K = 7$. The results can be read as the probability of a genus to be found in a set of prototypical locations. Interestingly, `Dir-Dir GS` is the method that returns the most detailed dictionary for this dataset. The other methods tend to produce larger clusters of genera. This highlights the importance of choosing the right model for each dataset since they imply different underlying assumptions. `Dir-Dir GS` assumes one topic per genus and one topic per location.

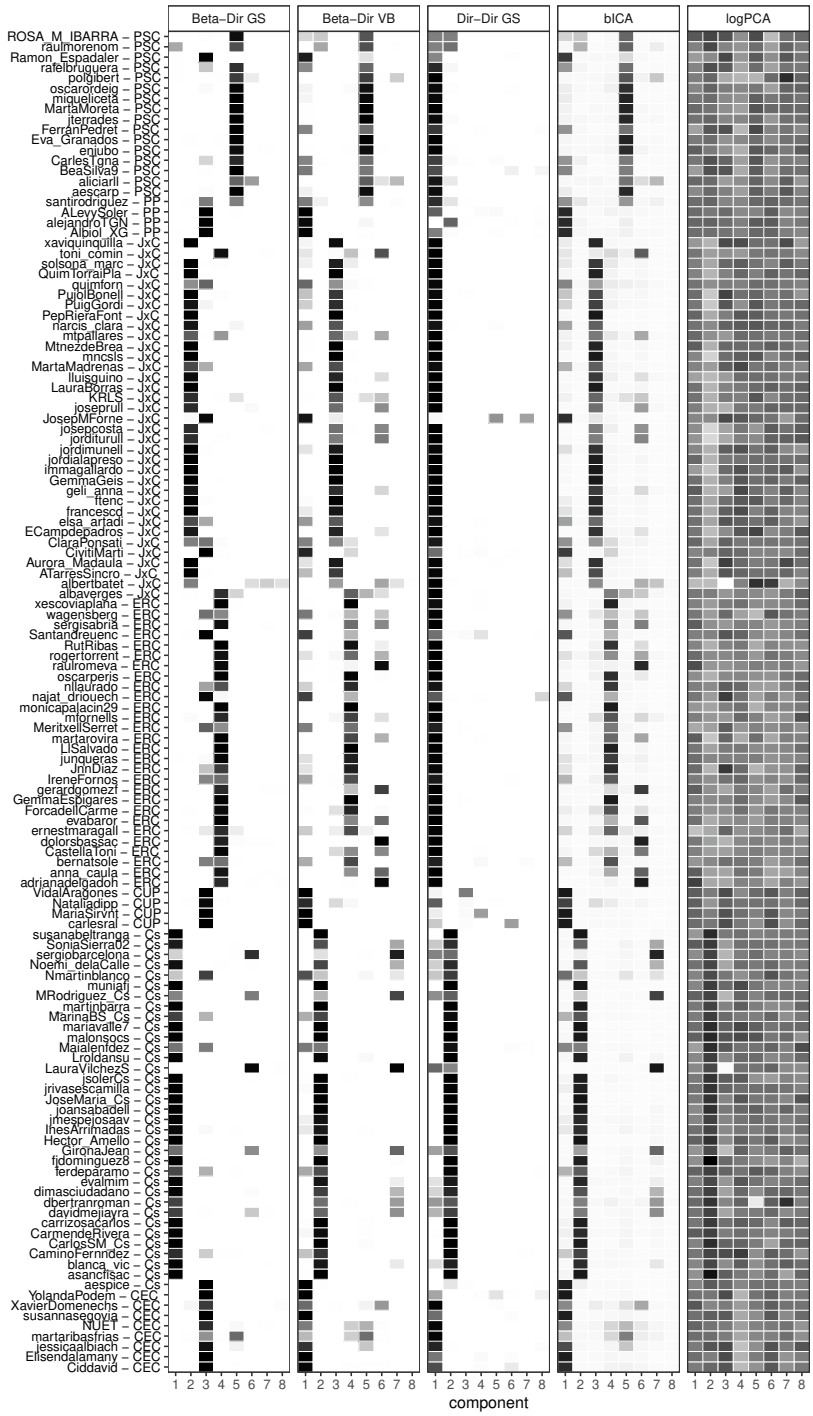


Figure 7 Estimated dictionaries from the `parliament` dataset. Members are sorted by party and then alphabetically. Columns are sorted by their norm. Only the first eight columns are displayed for the nonparametric methods `Beta-Dir GS`, `Beta-Dir VB` and `Dir-Dir GS`. The results displayed for `bICA` and `logPCA` are with $K = 8$. The values of $\mathbf{W}$ estimated by `logPCA-8` belong to the $[-207.17, 217.92]$ range and have been linearly mapped to the $[0, 1]$ range for visual display.

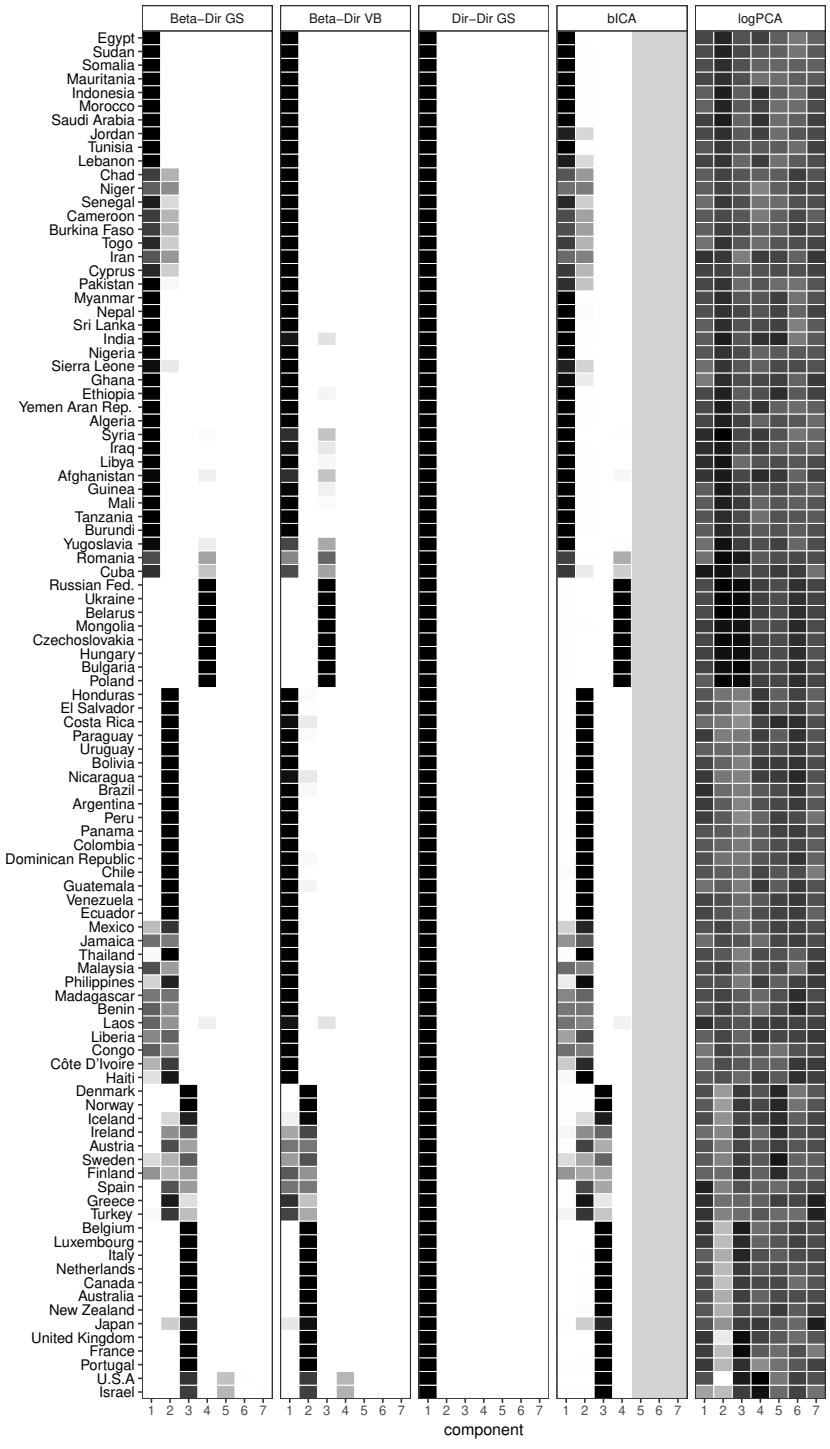


Figure 8 Estimated dictionaries from the `unvotes` dataset. Columns are sorted by their norm. Only the first seven columns are displayed for the nonparametric methods `Beta-Dir GS`, `Beta-Dir VB` and `Dir-Dir GS`. The results displayed for `bICA` and `logPCA` are with $K = 4$ and $K = 7$, respectively. The values of $\mathbf{W}$ estimated by `logPCA-7` belong to the $[-629.7, 335.1]$ range and have been linearly mapped to the $[0, 1]$ range for visual display.

We will see in the following section that `Dir-Dir GS` also gives the best predictions for this dataset. Again, the dictionary obtained with `logPCA` is harder to interpret.

6.4 Prediction

6.4.1 Experimental setting

We now evaluate the capability of the five previously considered methods together with `c-bICA` to predict missing data. For each of the five considered datasets, we applied the algorithms to a 75% random subset of the original data. We here use the full `unvotes` in which abstentions are treated as negative votes ($v_{fn} = 0$). `bICA` and `c-bICA` where applied with $K = 2, \ldots, 8$. `logPCA` was applied with $K = 2, \ldots, 4$. Then we computed the *perplexity* of the test set (the 25% held-out entries) given the estimate $\hat{\mathbf{V}} = \mathbf{E}[\mathbf{WH}|\mathbf{V}_{\text{train}}]$ (for all methods except `logPCA-K`) or $\hat{\mathbf{V}} = \sigma(\hat{\mathbf{W}}\hat{\mathbf{H}})$ (for `logPCA`). The perplexity is here simply taken as the negative log-likelihood of the test set (Hofmann, 1999):

$$perplexity = -\frac{1}{T} \sum_{(f,n)\in \text{test}} \log p(v_{fn}|\hat{\mathbf{V}}) \tag{67}$$

where T is the number of elements in the test set (in our case, $T = 0.25FN$).

6.4.2 Prediction performance

Fig. 10 displays the perplexities obtained by all methods from 10 repetitions of the experiment with randomly selected training and test sets, and random initializations (the same starting point is used for `bICA` and `c-bICA`). The proposed `Beta-Dir VB` performs similarly or better (`lastfm`, `parliament`) than `bICA`, while automatically adjusting the number of relevant components. As hinted from the dictionary learning experiments, `Dir-Dir GS` performs considerably better than the other mean-parameterized methods on the `paleo` dataset. `c-bICA` does not specifically improve over `bICA` (remember they are based on the same model, only inference changes) and performs worse in some cases (`lastfm`, `parliament`). However, its performance is more stable, with less variation between different runs, a likely consequence of the collapsed inference.

Despite its flexibility (unconstrained $\mathbf{W}$ and $\mathbf{H}$), `logPCA` provides marginally better perplexity (except on the `animal` dataset where it performs worse than almost all other methods), and only given a suitable value of K. Its predictive performance can drastically decay with ill-chosen values of K. In contrast, our proposed methods do not require tuning K to a proper value. Furthermore, they provide competitive prediction performance together with the interpretability of the decomposition. We have also compared against a standard `LDA` using the same range of K than `logPCA`, but the perplexity was much worse (around six) for all datasets; we did not plot it due to the high difference in the scale.

6.4.3 Convergence of the variational inference algorithms

Fig. 11 displays the average perplexity values returned by the variational algorithms `Beta-Dir VB`, `c-bICA-5` and `bICA-5` along iterations. As expected, `c-bICA` tends to converge faster than `bICA` though not consistently so. Being initialized with a full tensor of dimension $K = 100$ (as described in Section 6.2), `Beta-Dir VB` starts with a relatively higher perplexity but catches up with the two other methods in a reasonable number of iterations.

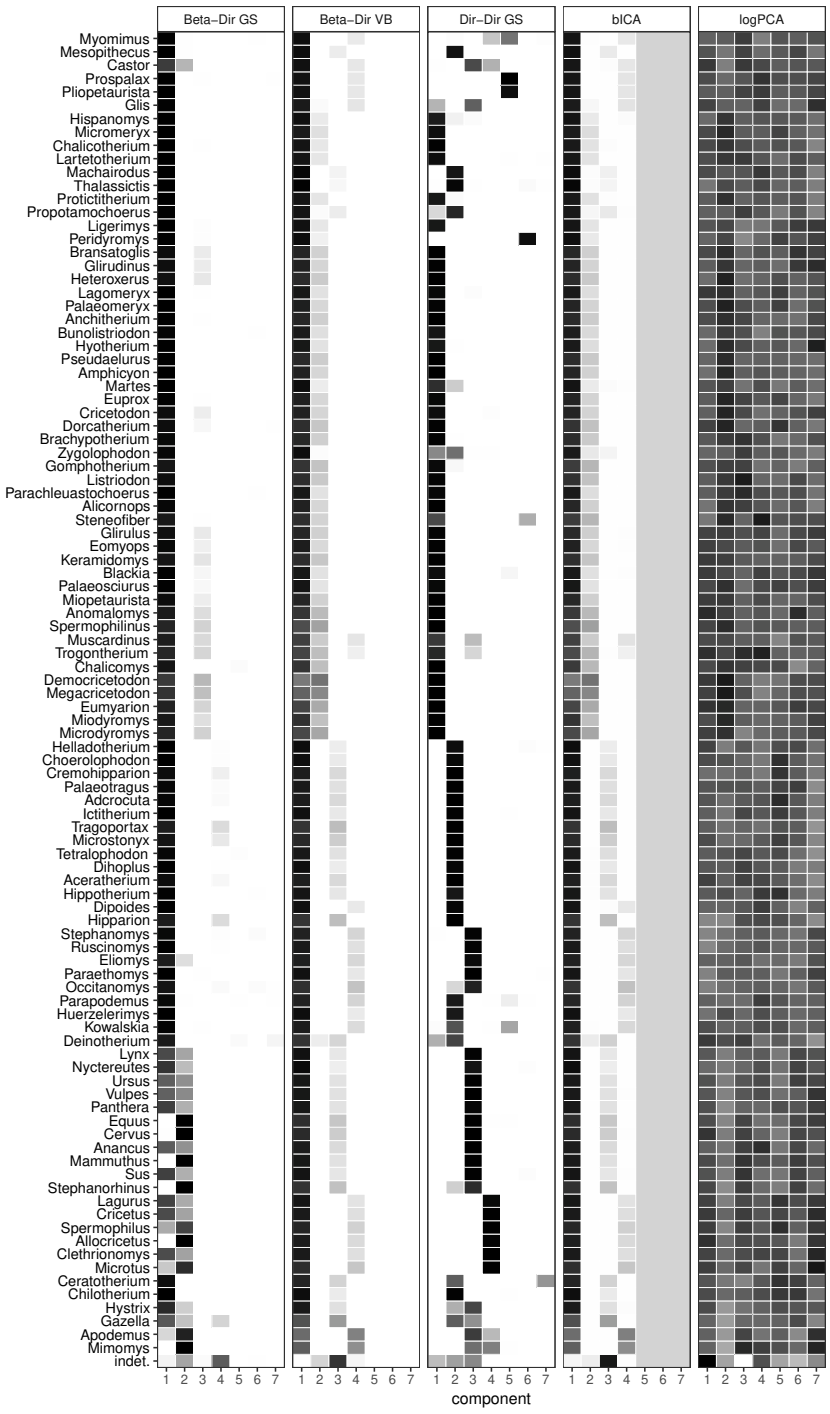


Figure 9 Estimated dictionaries from the `paleo` dataset. Columns are sorted by their norm. Only the first seven columns are displayed for the nonparametric methods `Beta-Dir GS`, `Beta-Dir VB` and `Dir-Dir GS`. The results displayed for `bICA` and `logPCA` are with $K = 4$ and $K = 7$, respectively. The values of $\mathbf{W}$ estimated by `logPCA-7` belong to the $[-102.3, 118.7]$ range and have been linearly mapped to the $[0, 1]$ range for visual display.

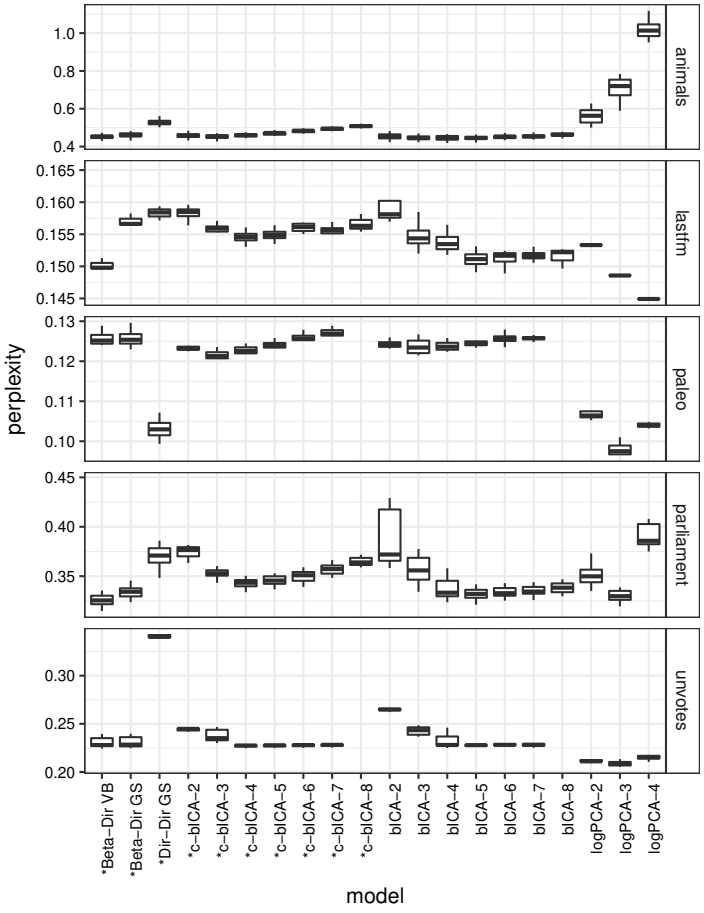


Figure 10 Prediction performance measured by perplexity (lower values are better). The methods introduced in this paper are marked with an asterisk.

7 Conclusions

We have presented a unified view for Bayesian mean-parameterized NBMF. The interest of mean-parameterized models in NMF is that they keep factors interpretable since they belong to the same space than the observed data. We have addressed three models that correspond to three possible sets of constraints that each respect mean-parameterization. One model, `Dir-Beta`, is a Bayesian extension of the Aspect Bernoulli model of Bingham et al. (2009). Another model, `Beta-Dir`, corresponds to the binary ICA model of Kabán and Bingham (2008). We have proposed a new collapsed Gibbs sampler and a new collapsed variational inference method for estimation in these models. We have proposed a novel, third model, `Dir-Dir`, and we have designed a collapsed Gibbs sampler for inference with this model. Lastly, we have proposed a nonparametric extension for these three models. Experiments have shown that our nonparametric methods can achieve similar performance than the state-of-the-art methods applied with a suitable value of K. As expected, the more flexible `logPCA` can achieve better

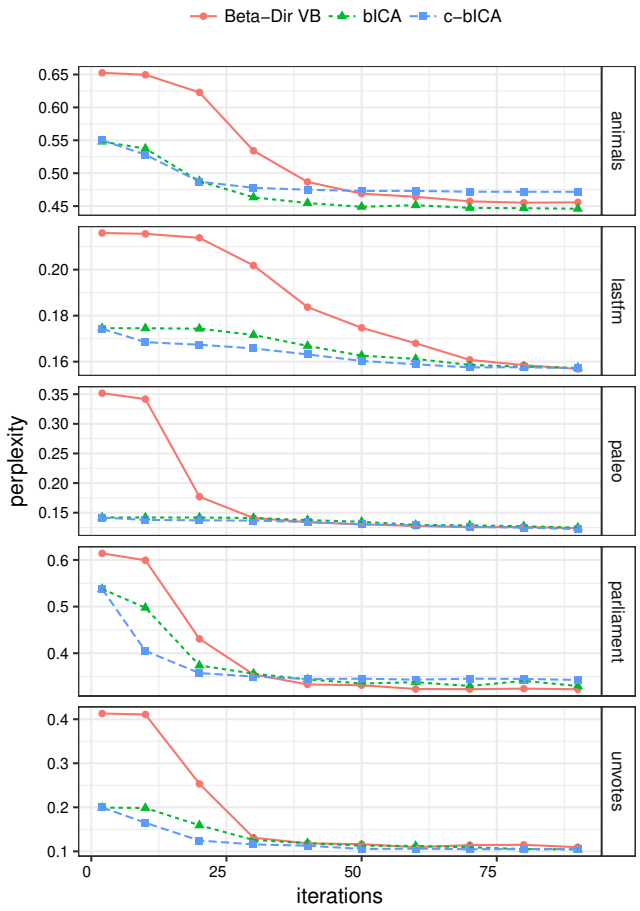


Figure 11 Average perplexity (over 10 repetitions) returned by the variational inference algorithms along iterations.

data approximation and in some cases prediction, but at the cost of interpretation which of utter importance in some applications.

Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 681839 (project FACTORY).

References

Aldous DJ (1985) Exchangeability and related topics. In: École d'été de probabilités de Saint-Flour, XIII—1983, Lecture Notes in Mathematics, vol 1117, Springer, Berlin, pp 1–198

Alquier P, Guedj B (2017) An oracle inequality for quasi-bayesian nonnegative matrix factorization. Mathematical Methods of Statistics 26(1):55–67

Anderson JR (1991) The adaptive nature of human categorization. Psychological Review 98:409–429

Asuncion A, Welling M, Smyth P, Teh YW (2009) On smoothing and inference for topic models. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09), pp 27–34

Bingham E, Kabán A, Fortelius M (2009) The Aspect Bernoulli model: Multiple causes of presences and absences. Pattern Analysis and Applications 12(1):55–78

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. Journal of Machine Learning Research 3(Jan):993–1022

Buntine WL, Jakulin A (2006) Discrete component analysis. In: Lecture Notes in Computer Science, Springer, vol 3940, pp 1–33

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 (SIGIR '04), pp 122–129

Çapan G, Akbayrak S, Ceritli TY, Cemgil AT (2018) Sum conditioned Poisson factorization. In: Proceedings of the 14th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA '18), pp 24–35

Celma O (2010) Music recommendation and discovery in the long tail. Springer

Cemgil AT (2009) Bayesian inference for nonnegative matrix factorisation models. Computational Intelligence and Neuroscience 2009:1–17

Cichocki A, Lee H, Kim YD, Choi S (2008) Non-negative matrix factorization with α-divergence. Pattern Recognition Letters 29(9):1433–1440

Collins M, Dasgupta S, Schapire RE (2002) A generalization of principal components analysis to the exponential family. In: Advances in Neural Information Processing Systems 14, MIT Press, pp 617–624

Févotte C, Idier J (2011) Algorithms for nonnegative matrix factorization with the β-divergence. Neural Computation 23(9):2421–2456

Gopalan P, Ruiz FJR, Ranganath R, Blei DM (2014) Bayesian nonparametric Poisson factorization for recommendation systems. Journal of Machine Learning Research 33:275–283

Gopalan P, Hofman JM, Blei DM (2015) Scalable recommendation with hierarchical Poisson factorization. In: Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI '15), pp 326–335

He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web (WWW '17), International World Wide Web Conferences Steering Committee, p 173–182

Hernandez-Lobato JM, Houlsby N, Ghahramani Z (2014) Stochastic inference for scalable probabilistic modeling of binary matrices. Proceedings of the 31st International Conference on Machine Learning (ICML '14) 32:1–6

Hoffman MD, Blei DM, Cook PR (2010) Bayesian nonparametric matrix factorization for recorded music. In: Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML '10), pp 439–446

Hoffman MD, Blei DM, Wang C, Paisley J (2013) Stochastic variational inference. Journal of Machine Learning Research 14(1):1303–1347

Hofmann T (1999) Probabilistic latent semantic analysis. In: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI '99), pp 289–296

Ishiguro K, Sato I, Ueda N (2017) Averaged collapsed variational Bayes inference. Journal of Machine Learning Research 18(1):1–29

Kabán A, Bingham E (2008) Factorisation and denoising of 0-1 data: A variational approach. Neurocomputing 71(10-12):2291–2308

Kemp C, Tenenbaum JB, Griffiths TL, Yamada T, Ueda N (2006) Learning systems of concepts with an infinite relational model. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06), pp 381–388

Landgraf AJ, Lee Y (2015) Dimensionality reduction for binary data through the projection of natural parameters. Tech. Rep. 890, Department of Statistics, The Ohio State University

Larsen JS, Clemmensen LKH (2015) Non-negative matrix factorization for binary data. In: Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K '15), pp 555–563

Lee DD, Seung HS (1999) Learning the parts of objects with nonnegative matrix factorization. Nature 401:788–791

Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems 13, pp 556–562

Liu JS (1994) The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. Journal of the American Statistical Association 89(427):958–966

Meeds E, Ghahramani Z, Neal RM, Roweis ST (2007) Modeling dyadic data with binary latent factors. In: Advances in Neural Information Processing Systems 19, MIT Press, pp 977–984

Miettinen P, Mielikäinen T, Gionis A, Das G, Mannila H (2008) The discrete basis problem. IEEE Transactions on Knowledge and Data Engineering 20(10):1348–1362

NOW (2018) The NOW community. new and old worlds database of fossil mammals (NOW). URL http://www.helsinki.fi/science/now/, release 030717, retrieved May 2018

Pitman J (2002) Combinatorial stochastic processes. Lecture Notes in Mathematics, Springer-Verlag, Berlin, lectures from the 32nd Summer School on Probability Theory held in Saint-Flour

Polson NG, Scott JG, Windle J (2013) Bayesian inference for logistic models using Polya-Gamma latent variables. Journal of the American Statistical Association 108:1339–1349

R Core Team (2017) R: A language and environment for statistical computing. R Foundation for Statistical Computing, URL https://www.R-project.org/

Rukat T, Holmes CC, Titsias MK, Yau C (2017) Bayesian Boolean matrix factorisation. In: Proceedings of the 34th International Conference on Machine Learning (ICML '17), pp 2969–2978

Sammel MD, Ryan LM, Legler JM (1997) Latent variable models for mixed discrete and continuous outcomes. Journal of the Royal Statistical Society Series B (Methodological) 59(3):667–678

Schein AI, Saul LK, Ungar LH (2003) A generalized linear model for principal component analysis of binary data. In: Proceedings of the 9th Workshop on Artificial Intelligence and Statistics (AISTATS '03), pp 14–21

Schmidt MN, Winther O, Hansen LK (2009) Bayesian non-negative matrix factorization. In: Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation (ICA '09), pp 540–547

Singh AP, Gordon GJ (2008) A unified view of matrix factorization models. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD '08), pp 358–373

Slawski M, Hein M, Lutsik P (2013) Matrix factorization with binary components. In: Advances in Neural Information Processing Systems 26, Curran Associates, Inc., pp 3210–3218

Sørensen T (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Biologiske Skrifter 5:1–34

Steck H, Jaakkola TS (2003) On the Dirichlet prior and Bayesian regularization. In: Advances in Neural Information Processing Systems 15, MIT Press, pp 713–720

Teh YW, Jordan MI, Beal MJ, Blei DM (2006) Hierarchical Dirichlet processes. Journal of the American Statistical Association 101(476):1566–1581

Teh YW, Newman D, Welling M (2007) A collapsed variational Bayesian inference alzgorithm for latent Dirichlet allocation. In: Advances in Neural Information Processing Systems 19, MIT Press, pp 1353–1360

Tipping ME (1999) Probabilistic visualisation of high-dimensional binary data. In: Advances in Neural Information Processing Systems 11, MIT Press, pp 592–598

Tipping ME, Bishop C (1999) Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B 21/3:611–622

Tomé AM, Schachtner R, Vigneron V, Puntonet CG, Lang EW (2013) A logistic non-negative matrix factorization approach to binary data sets. Multidimensional Systems and Signal Processing 26(1):125–143

Udell M, Horn C, Zadeh R, Boyd S (2016) Generalized low rank models. Foundations and Trends in Machine Learning 9(1):1–118

Voeten E (2013) Data and analyses of voting in the UN General Assembly. In: Reinal B (ed) Routledge Handbook of International Organization, Routledge

Xue HJ, Dai XY, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI '17), AAAI Press, p 3203–3209

Zhang ZY, Li T, Ding C, Ren XW, Zhang XS (2009) Binary matrix factorization for analyzing gene expression data. Data Mining and Knowledge Discovery 20(1):28–52

Zhou M (2015) Infinite edge partition models for overlapping community detection and link prediction. In: Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS '15), pp 1135–1143

Zhou M, Hannah L, Dunson D, Carin L (2012) Beta-negative binomial process and Poisson factor analysis. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS '12), pp 1462–1471

Zhou M, Cong Y, Chen B (2016) Augmentable Gamma belief networks. Journal of Machine Learning Research 17(163):1–44

A Probability distributions functions

A.1 Bernoulli distribution

Distribution over a binary variable $x \in \{0, 1\}$, with mean parameter $\mu \in [0, 1]$:

$$\text{Bernoulli}(x|\mu) = \mu^x (1-\mu)^{1-x}. \tag{68}$$

A.2 Beta distribution

Distribution over a continuous variable $x \in [0, 1]$, with shape parameters $a > 0$, $b > 0$:

$$\text{Beta}(x|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}. \tag{69}$$

A.3 Gamma distribution

Distribution for a continuous variable $x > 0$, with shape parameter $a > 0$ and rate parameter $b > 0$:

$$\text{Gamma}(x|a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}. \tag{70}$$

A.4 Dirichlet distribution

Distribution for K continuous variables $x_k \in [0, 1]$ such that $\sum_k x_k = 1$. Governed by K shape parameters $\alpha_1, ...\alpha_K$ such that $\alpha_k > 0$:

$$\text{Dirichlet}(\mathbf{x}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k x^{\alpha_k - 1}. \tag{71}$$

A.5 Discrete distribution

Distribution for the discrete variable $\mathbf{x} \in \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$, where $\mathbf{e}_i$ is the i^{th} canonical vector. Governed by the discrete probabilities $\mu_1, ..., \mu_K$ such that $\mu_k \in [0, 1]$ and $\sum_k \mu_k = 1$:

$$p(\mathbf{x} = \mathbf{e}_k) = \mu_k \tag{72}$$

The probability mass function can be written as:

$$\text{Discrete}(\mathbf{x}|\boldsymbol{\mu}) = \prod_k \mu_k^{x_k}. \tag{73}$$

We may write $\text{Discrete}(\mathbf{x}|\boldsymbol{\mu}) = \text{Multinomial}(\mathbf{x}|1, \boldsymbol{\mu})$.

A.6 Multinomial distribution

Distribution for an integer-valued vector $\mathbf{x} = [x_1, ..., x_K]^T \in \mathbb{N}^K$. Governed by the total number $L = \sum_k x_k$ of events assigned to K bins and the probabilities μ_k of being assigned to bin k:

$$\text{Multinomial}(\mathbf{x}|L, \boldsymbol{\mu}) = \frac{L!}{x1!...x_K!} \prod_k \mu_k^{x_k}. \tag{74}$$

B Derivations for the `Beta-Dir` model

B.1 Marginalizing out W and H from the joint likelihood

We seek to compute marginal joint probability introduced in Eq. (22) and given by:

$$p(\mathbf{V}, \mathbf{Z}) = \prod_f \overbrace{\int p(\mathbf{w}_f) \prod_n p(\mathbf{z}_{fn}|\mathbf{w}_f)\, d\mathbf{w}_f}^{p(\underline{\mathbf{Z}}_f)} \prod_n \overbrace{\int \prod_k p(h_{kn}) \prod_f p(v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn})\, d\mathbf{h}_n}^{p(\mathbf{v}_n|\mathbf{Z}_n)}.$$

Using the expression of the normalization constant of the Dirichlet distribution, the first integral can be computed as follows:

$$\begin{aligned}
p(\underline{\mathbf{Z}}_f) &= \int p(\mathbf{w}_f) \prod_n p(\mathbf{z}_{fn}|\mathbf{w}_f)\, d\mathbf{w}_f && (75)\\
&= \int \frac{\Gamma(\sum_k \gamma_k)}{\prod_k \Gamma(\gamma_k)} \prod_k w_{fk}^{\gamma_k - 1} \prod_n w_{fk}^{z_{fkn}}\, d\mathbf{w}_f && (76)\\
&= \frac{\Gamma(\sum_k \gamma_k)}{\prod_k \Gamma(\gamma_k)} \int \prod_k w_{fk}^{\gamma_k + L_{fk} - 1}\, d\mathbf{w}_f && (77)\\
&= \frac{\Gamma(\sum_k \gamma_k)}{\prod_k \Gamma(\gamma_k)} \frac{\prod_k \Gamma(\gamma_k + L_{fk})}{\Gamma(\sum_k \gamma_k + L_{fk})}. && (78)
\end{aligned}$$

The second integral in Eq. (22) is computed as follows. In Eq. (80) we use that $p(v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn}) = \text{Bernoulli}(v_{fn}|\prod_k h_{kn}^{z_{fkn}}) = \prod_k \text{Bernoulli}(v_{fn}|h_{kn})^{z_{fkn}}$ (recall that $\mathbf{z}_{fn}$ is an indicator vector).

In Eq. (83), we use the expression of the normalization constant of the Beta distribution.

$$p(\mathbf{v}_n|\mathbf{Z}_n) = \int \prod_k p(h_{kn}) \prod_f p(v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn})\, d\mathbf{h}_n \tag{79}$$

$$= \int \prod_k \left[\frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} h_{kn}^{\alpha_k-1}(1-h_{kn})^{\beta_k-1}\right] \prod_{fk} \left[h_{kn}^{v_{fn}}(1-h_{kn})^{1-v_{fn}}\right]^{z_{fkn}} d\mathbf{h}_n \tag{80}$$

$$= \prod_k \int \frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} h_{kn}^{\alpha_k-1}(1-h_{kn})^{\beta_k-1} \prod_f \left[h_{kn}^{v_{fn}}(1-h_{kn})^{1-v_{fn}}\right]^{z_{fkn}} dh_{kn} \tag{81}$$

$$= \prod_k \frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \int h_{kn}^{\alpha_k+A_{kn}-1}(1-h_{kn})^{\beta_k+B_{kn}-1} dh_{kn} \tag{82}$$

$$= \prod_k \frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \frac{\Gamma(\alpha_k+A_{kn})\Gamma(\beta_k+B_{kn})}{\Gamma(\alpha_k+\beta_k+M_{kn})}. \tag{83}$$

B.2 Conditional prior and posterior distributions of $\mathbf{z}_{fn}$

Applying the Bayes rule, the conditional posterior of $\mathbf{z}_{fn}$ is given by:

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{V}) \propto p(\mathbf{V}|\mathbf{Z}) p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}). \tag{84}$$

The likelihood itself decomposes as $p(\mathbf{V}|\mathbf{Z}) = \prod_n p(\mathbf{v}_n|\mathbf{Z}_n)$ and we may ignore the terms that do not depend on $\mathbf{z}_{fn}$. Using Eq. (24) and the identity $\Gamma(n+b) = \Gamma(n)n^b$ where b is a binary variable, we may write:

$$p(\mathbf{v}_n|\mathbf{Z}_n) = \prod_k \frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \frac{\Gamma(\alpha_k+A_{kn})\Gamma(\beta_k+B_{kn})}{\Gamma(\alpha_k+\beta_k+M_{kn})} \tag{85}$$

$$\propto \prod_k \frac{\Gamma(\alpha_k+A_{kn})\Gamma(\beta_k+B_{kn})}{\Gamma(\alpha_k+\beta_k+M_{kn})} \tag{86}$$

$$= \prod_k \frac{\Gamma(\alpha_k+A_{kn}^{\neg fn}+z_{fkn}v_{fn})\Gamma(\beta_k+B_{kn}^{\neg fn}+z_{fkn}\bar{v}_{fn})}{\Gamma(\alpha_k+\beta_k+M_{kn}^{\neg fn}+z_{fkn})} \tag{87}$$

$$\propto \prod_k \frac{\Gamma(\alpha_k+A_{kn}^{\neg fn})(\alpha_k+A_{kn}^{\neg fn})^{z_{fkn}v_{fn}}\Gamma(\beta_k+B_{kn}^{\neg fn})(\beta_k+B_{kn}^{\neg fn})^{z_{fkn}\bar{v}_{fn}}}{\Gamma(\alpha_k+\beta_k+M_{kn}^{\neg fn})(\alpha_k+\beta_k+M_{kn}^{\neg fn})^{z_{fkn}}} \tag{88}$$

$$\propto \prod_k \left[\frac{(\alpha_k+A_{kn}^{\neg fn})^{v_{fn}}(\beta_k+B_{kn}^{\neg fn})^{\bar{v}_{fn}}}{(\alpha_k+\beta_k+M_{kn}^{\neg fn})}\right]^{z_{fkn}}. \tag{89}$$

The conditional prior term is given by

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}) = p(\mathbf{Z})/p(\mathbf{Z}_{\neg fn}). \tag{90}$$

Using $p(\mathbf{Z}) = \prod_f \underline{\mathbf{Z}}_f$ and Eq. (23) we have

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}) \propto p(\underline{\mathbf{Z}}_f) \tag{91}$$

$$\propto \prod_k \Gamma(\gamma_k + L_{kn}^{\neg fn} + z_{fkn}) \tag{92}$$

$$= \prod_k \Gamma(\gamma_k + L_{kn}^{\neg fn})(\gamma_k + L_{kn}^{\neg fn})^{z_{fkn}} \tag{93}$$

$$\propto \prod_k (\gamma_k + L_{kn}^{\neg fn})^{z_{fkn}}. \tag{94}$$

Using $\sum_k p(\mathbf{z}_{fn} = \mathbf{e}_k|\mathbf{Z}_{\neg fn}) = 1$, a simple closed-form expression of $p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn})$ is obtained as follows:

$$p(\mathbf{z}_{fn} = \mathbf{e}_k|\mathbf{Z}_{\neg fn}) = \frac{\gamma_k + L_{kn}^{\neg fn}}{\sum_k (\gamma_k + L_{kn}^{\neg fn})} \tag{95}$$

$$= \frac{\gamma_k + L_{kn}^{\neg fn}}{\sum_k \gamma_k + N - 1}. \tag{96}$$

Combining Eqs. (84), (89) and (94), we obtain

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{V}) \propto \prod_k \left[(\gamma_k + L_{fk}^{\neg fn}) \frac{(\alpha_k + A_{kn}^{\neg fn})^{v_{fn}} (\beta_k + B_{kn}^{\neg fn})^{\bar{v}_{fn}}}{\alpha_k + \beta_k + M_{kn}^{\neg fn}} \right]^{z_{fkn}}. \tag{97}$$

C Alternative Gibbs sampler for the Dir-Dir model

In this appendix, we show how to derive an alternative Gibbs sampler based on a single augmentation, like in the `Beta-Dir` model. This is a conceptually interesting result, though it does not lead to an efficient implementation.

Likewise the `Beta-Dir` model, the `Dir-Dir` model can be augmented using the single indicator variables $\mathbf{z}_{fn}$, as follows:

$$\mathbf{h}_n \sim \text{Dirichlet}(\boldsymbol{\eta}) \tag{98}$$

$$\mathbf{w}_f \sim \text{Dirichlet}(\boldsymbol{\gamma}) \tag{99}$$

$$\mathbf{z}_{fn}|\mathbf{w}_f \sim \text{Discrete}(\mathbf{w}_f) \tag{100}$$

$$v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn} \sim \text{Bernoulli}\left(\prod_k h_{kn}^{z_{fkn}} \right) \tag{101}$$

Note that compared to Eqs. (15)-(18) only the prior on $\mathbf{h}_n$ is changed.

Like in `Beta-Dir`, we seek in this appendix to derive a Gibbs sampler from the conditional probabilities $p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{V})$ given by

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}, \mathbf{V}) \propto p(\mathbf{V}|\mathbf{Z}) p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}). \tag{102}$$

The conditional prior term is identical to that of `Beta-Dir` and given by

$$p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn}) \propto \prod_k (\gamma_k + L_{kn}^{\neg fn})^{z_{fkn}}. \tag{103}$$

Like in `Beta-Dir`, the likelihood term factorizes as $p(\mathbf{V}|\mathbf{Z}) = \prod_n p(\mathbf{v}_n|\mathbf{Z}_n)$, and we now derive the expression of $p(\mathbf{v}_n|\mathbf{Z}_n)$. As compared to `Beta-Dir`, a major source of difficulty lies in the fact that $p(\mathbf{h}_n)$ does not fully factorize anymore because of the Dirichlet assumption (and in particular $\sum_k h_{kn} = 1$). In the following, we use the multinomial theorem to obtain Eq. (107)[4] and we use the expression of the normalization constant of the Dirichlet distribution to obtain Eq. (110):

$$p(\mathbf{v}_n|\mathbf{Z}_n) = \int p(\mathbf{h}_n) \prod_f p(v_{fn}|\mathbf{h}_n, \mathbf{z}_{fn})\, d\mathbf{h}_n \tag{104}$$

$$= \int \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \prod_k h_{kn}^{\eta_k - 1} \prod_f \prod_k \left[h_{kn}^{v_{fn}} (1 - h_{kn})^{1 - v_{fn}}\right]^{z_{fkn}} d\mathbf{h}_n \tag{105}$$

$$= \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \int \prod_k h_{kn}^{\eta_n + A_{kn} - 1} (1 - h_{kn})^{B_{kn}}\, d\mathbf{h}_n \tag{106}$$

$$= \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \int \prod_k h_{kn}^{\eta_n + A_{kn} - 1} \sum_{j_k=0}^{B_{kn}} \binom{B_{kn}}{j_k} (-h_{kn})^{j_k}\, d\mathbf{h}_n \tag{107}$$

$$= \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \int \sum_{j_1=0}^{B_{1n}} \cdots \sum_{j_K=0}^{B_{Kn}} \prod_k h_{kn}^{\eta_k + A_{kn} - 1} \binom{B_{kn}}{j_k} (-h_{kn})^{j_k}\, d\mathbf{h}_n \tag{108}$$

$$= \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \sum_{j_1=0}^{B_{1n}} \cdots \sum_{j_K=0}^{B_{Kn}} \prod_k (-1)^{j_k} \binom{B_{kn}}{j_k} \int \prod_k h_{kn}^{\eta_k + A_{kn} + j_k - 1}\, d\mathbf{h}_n \tag{109}$$

$$= \frac{\Gamma(\sum_k \eta_k)}{\prod_k \Gamma(\eta_k)} \sum_{j_1=0}^{B_{1n}} \cdots \sum_{j_K=0}^{B_{Kn}} \prod_k (-1)^{j_k} \binom{B_{kn}}{j_k} \frac{\Gamma(\eta_k + A_{kn} + j_k)}{\Gamma(\sum_k \eta_k + A_{kn} + j_k)}. \tag{110}$$

We conclude that, though available in closed form, the expression of $p(\mathbf{v}_n|\mathbf{Z}_n)$ (and thus $p(\mathbf{z}_{fn}|\mathbf{Z}_{\neg fn})$) involves the computation of $K \prod_{k=1}^{K} B_{kn}$ terms involving binomial coefficients, which is impractical in typical problem dimensions.

[4] Many thanks to Xi'an (Christian Robert) for giving us the trick via StackExchange.