Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2009, Article ID 785152, 17 pages
doi:10.1155/2009/785152

Research Article

Bayesian Inference for Nonnegative Matrix Factorisation Models

Ali Taylan Cemgil

Department of Computer Engineering, Boğaziçi University, 34342 Bebek, Istanbul, Turkey

Correspondence should be addressed to Ali Taylan Cemgil, atc27@cam.ac.uk

Received 29 August 2008; Accepted 14 February 2009

Recommended by S. Cruces-Alvarez

We describe nonnegative matrix factorisation (NMF) with a Kullback-Leibler (KL) error measure in a statistical framework, with a hierarchical generative model consisting of an observation and a prior component. Omitting the prior leads to the standard KL-NMF algorithms as special cases, where maximum likelihood parameter estimation is carried out via the Expectation-Maximisation (EM) algorithm. Starting from this view, we develop full Bayesian inference via variational Bayes or Monte Carlo. Our construction retains conjugacy and enables us to develop more powerful models while retaining attractive features of standard NMF such as monotonic convergence and easy implementation. We illustrate our approach on model order selection and image reconstruction.

1. Introduction

In machine learning, nonnegative matrix factorisation (NMF) was introduced by Lee and Seung [1] as an alternative to k-means clustering and principal component analysis (PCA) for data analysis and compression (also see [2]). In NMF, given a $W \times K$ nonnegative matrix $X = \{x_{\nu,\tau}\}$, where $\nu = 1:W$, $\tau = 1:K$, we seek positive matrices T and V such that

$$x_{\nu,\tau} \approx [TV]_{\nu,\tau} = \sum_i t_{\nu,i} v_{i,\tau}, \tag{1}$$

where $i = 1:I$. We will refer to the $W \times I$ matrix T as the *template matrix*, and $I \times K$ matrix V the *excitation matrix*. The key property of NMF is that T and V are constrained to be positive matrices. This is in contrast with PCA, where there are no positivity constraints or k-means clustering where each column of V is constrained to be a unit vector. Subject to the positivity constraints, we seek a solution to the following minimisation problem:

$$(T,V)^* = \arg\min_{T,V>0} D(X\|TV). \tag{2}$$

Here, the function D is a suitably chosen error function. One particular choice for D, on which we will focus here, is the information (Kullback-Leibler) divergence, which we write as

$$D(X\|\Lambda) = -\sum_{\nu,\tau}\left(x_{\nu,\tau}\log\frac{\lambda_{\nu,\tau}}{x_{\nu,\tau}} - \lambda_{\nu,\tau} + x_{\nu,\tau}\right). \tag{3}$$

Using Jensen's inequality [3] and concavity of $\log x$, it can be shown that $D(\cdot)$ is nonnegative and $D(X||\Lambda) = 0$ if and only if $X = \Lambda$. The objective in (2) could be minimised by any suitable optimisation algorithm. Lee and Seung [1] have proposed a very efficient variational bound minimisation algorithm that has attractive convergence properties and which has been successfully applied in various applications in signal analysis and source separation, for example, [4–6].

The interpretation of NMF, like singular value decomposition (SVD), as a low rank matrix approximation is sufficient for the derivation of a useful inference algorithm; yet this view arguably does not provide the complete picture about assumptions underlying the statistical properties of X. Therefore, we describe NMF from a statistical perspective as a hierarchical model. In our framework, the original nonnegative multiplicative update equations of NMF appear as an expectation-maximisation (EM) algorithm for maximum likelihood estimation of a conditionally Poisson model via *data augmentation*. Starting from this view, we develop Bayesian extensions that facilitate more powerful modelling and allow more sophisticated inference, such as Bayesian

model selection. Inference in the resulting models can be carried out easily using variational (structured mean field) or Markov Chain Monte Carlo (Gibbs sampler). The resulting algorithms outperform existing NMF strategies and open up the way for a full Bayesian treatment for model selection via computation of the marginal likelihoods (the evidence), such as estimating the dimensions of the template matrix or regularising overcomplete representations via automatic relevance determination.

2. The Statistical Perspective

The interpretation of NMF as a low-rank matrix approximation is sufficient for the derivation of an inference algorithm; yet this view arguably does not provide the complete picture. In this section, we describe NMF from a statistical perspective. This view will pave the way for developing extensions that facilitate more realistic and flexible modelling as well as more sophisticated inference, such as Bayesian model selection.

Our first step is the derivation of the information divergence error measure from a maximum likelihood principle. We consider the following hierarchical model:

$$T \sim p(T \mid \Theta^t), \qquad V \sim p(V \mid \Theta^v), \tag{4}$$

$$s_{\nu,i,\tau} \sim \mathcal{PO}(s_{\nu,i,\tau}; t_{\nu,i} v_{i,\tau}), \qquad x_{\nu,\tau} = \sum_i s_{\nu,i,\tau}. \tag{5}$$

Here, $\mathcal{PO}(s;\lambda)$ denotes the Poisson distribution of the random variable $s \in \mathbb{N}_0$ with nonnegative intensity parameter λ, where

$$\mathcal{PO}(s;\lambda) = \exp(s \log \lambda - \lambda - \log \Gamma(s+1)) \tag{6}$$

and $\Gamma(s+1) = s!$ is the gamma function. The priors $p(T \mid \cdot)$ and $p(V \mid \cdot)$ will be specified later. We call the variables $S_i = \{s_{\nu,i,\tau}\}$ *latent sources*. We can analytically marginalise out the latent sources $S = \{S_1 \cdots S_I\}$ to obtain the marginal likelihood

$$\begin{aligned} \log p(X \mid T, V) &= \log \sum_S p(X \mid S) p(S \mid T, V) \\ &= \log \prod_{\nu,\tau} \mathcal{PO}\left(x_{\nu,\tau}; \sum_i t_{\nu,i}, v_{i,\tau}\right) \\ &= \sum_\nu \sum_\tau (x_{\nu,\tau} \log [TV]_{\nu,\tau} - [TV]_{\nu,\tau} \\ &\qquad - \log \Gamma(x_{\nu,\tau}+1)). \end{aligned} \tag{7}$$

This result follows from the well-known *superposition property* of Poisson random variables [7], namely, when $s_i \sim \mathcal{PO}(s_i;\lambda_i)$ and $x = s_1 + s_2 + \cdots + s_I$, then the marginal probability is given by $p(x) = \mathcal{PO}(x; \sum_i \lambda_i)$. The maximisation of this objective in T and V is equivalent to the minimisation of the information divergence in (3). In the derivation of original NMF in [8], this objective is stated first; the S variables are introduced implicitly later during the optimisation on T and V. In the sequel, we show that this algorithm is actually equivalent to EM, ignoring the priors $p(T \mid \cdot)$ and $p(V \mid \cdot)$.

2.1. Maximum Likelihood and the EM Algorithm. The log-likelihood of the observed data X can be written as

$$\begin{aligned} \mathcal{L}_X(T,V) &\equiv \log \sum_S p(X \mid S) p(S \mid T, V) \\ &\geq \sum_S q(S) \log \frac{p(X, S \mid T, V)}{q(S)} \equiv \mathcal{B}_{\mathrm{EM}}[q], \end{aligned} \tag{8}$$

where $q(S)$ is an instrumental distribution, that is arbitrary provided that the sum on the right exists; q can only vanish at a particular S only when p does so. Note that this defines a lower bound to the log-likelihood. It can be shown via functional derivatives and imposing the normalisation condition $\sum_S q(S) = 1$ via Lagrange multipliers that the lower bound is tight for the exact posterior of the latent sources, that is,

$$\arg\max_{q(S)} \mathcal{B}_{\mathrm{EM}}[q] = p(S \mid X, T, V). \tag{9}$$

Hence the log-likelihood can be maximised iteratively as follows:

$$\begin{aligned} &\text{E Step} \quad q(S)^{(n)} = p(S \mid X, T^{(n-1)}, V^{(n-1)}), \\ &\text{M Step} \quad (T^{(n)}, V^{(n)}) = \arg\max_{T,V} \langle \log p(S, X \mid T, V) \rangle_{q(S)^{(n)}}. \end{aligned} \tag{10}$$

Here, $\langle f(x) \rangle_{p(x)} = \int p(x) f(x) dx$, the expectation of some function $f(x)$ with respect to $p(x)$. In the E step, we compute the posterior distribution of S. This defines a lower bound on the likelihood

$$\mathcal{B}^{(n)}(T, V \mid T^{(n-1)}, V^{(n-1)}) = \langle \log p(S, X \mid T, V) \rangle_{q(S)^{(n)}}. \tag{11}$$

For many models in the exponential family, which includes (5), the expectation on the right depends on the sufficient statistics of $q(S)^{(n)}$ and is readily available; in fact calculating $q(S)$ should be literally taken as calculating the sufficient statistics of $q(S)$. The lower bound is readily obtained as a function of these sufficient statistics, and maximisation in the M Step yields a fixed point equation.

2.1.1. The E Step. To derive the posterior of the latent sources, we observe that

$$p(S \mid X, T, V) = \frac{p(S, X \mid T, V)}{p(X \mid T, V)}. \tag{12}$$

For the model in (5), we have

$$\begin{aligned} &\log p(S, X \mid T, V) \\ &= \sum_\nu \sum_\tau \left(\sum_i (-t_{\nu,i} v_{i,\tau} + s_{\nu,i,\tau} \log(t_{\nu,i} v_{i,\tau}) - \log \Gamma(s_{\nu,i,\tau}+1)) \right. \\ &\qquad \left. + \log \delta\left(x_{\nu,\tau} - \sum_i s_{\nu,i,\tau}\right)\right). \end{aligned} \tag{13}$$

It follows from (5), (12), (13), and (7)

$$\begin{aligned}
&\log p(S \mid X, T, V) \\
&\quad = \sum_{\nu}\sum_{\tau}\left(\sum_{i}\left(s_{\nu,i,\tau}\log\left(\frac{t_{\nu,i}v_{i,\tau}}{\sum_{i'}t_{\nu,i'}v_{i',\tau}}\right) - \log\Gamma(s_{\nu,i,\tau}+1)\right)\right. \\
&\qquad\qquad \left. + \log\Gamma(x_{\nu,\tau}+1) + \log\delta\left(x_{\nu,\tau} - \sum_{i}s_{\nu,i,\tau}\right)\right) \\
&\quad = \sum_{\nu}\sum_{\tau}\log\mathcal{M}(s_{\nu,1,\tau},\ldots,s_{\nu,I,\tau}; x_{\nu,\tau}, p_{\nu,1,\tau},\ldots,p_{\nu,I,\tau}),
\end{aligned} \tag{14}$$

where $p_{\nu,i,\tau} \equiv t_{\nu,i}v_{i,\tau}/\sum_{i'}t_{\nu,i'}v_{i',\tau}$ are the cell probabilities. Here, $\mathcal{M}$ denotes a multinomial distribution defined by

$$\begin{aligned}
\mathcal{M}(\mathbf{s}; x, \mathbf{p}) &= \binom{x}{s_1\, s_2\, \cdots\, s_I} p_1^{s_1}p_2^{s_2}\cdots p_I^{s_I}\delta\left(x - \sum_{i}s_i\right) \\
&= \delta\left(x - \sum_{i}s_i\right)x!\prod_{i=1}^{I}\frac{p_i^{s_i}}{s_i!},
\end{aligned} \tag{15}$$

where $\mathbf{s} = \{s_1, s_2, \ldots, s_I\}$, $\mathbf{p} = \{p_1, p_2, \ldots, p_I\}$, and $p_1 + p_2 + \cdots + p_I = 1$. Here, p_i, $i = 1 \cdots I$ are the cell probabilities, and x is the index parameter where $s_1 + s_2 + \cdots + s_I = x$. The Kronecker delta function is defined by $\delta(x) = 1$ when $x = 0$, and $\delta(x) = 0$ otherwise. It is a standard result that the marginal mean is

$$\langle s_i \rangle = x p_i, \tag{16}$$

that is, the expected value of each source s_i is a fraction of the observation, where the fraction is given by the corresponding cell probability.

2.1.2. The M Step. It is indeed a good news that the posterior has an analytic form. Since now the M step can be calculated easily as follows:

$$\begin{aligned}
&\langle \log p(S, X \mid T, V)\rangle_{p(S|X,T,V)} \\
&= \sum_{\nu}\sum_{\tau}\left(\sum_{i}\left(-t_{\nu,i}v_{i,\tau} + \langle s_{\nu,i,\tau}\rangle \log\left(t_{\nu,i}v_{i,\tau}\right) - \langle\log\Gamma(s_{\nu,i,\tau}+1)\rangle\right)\right. \\
&\qquad \left. + \left\langle \log\delta\left(x_{\nu,\tau} - \sum_{i}s_{\nu,i,\tau}\right)\right\rangle\right).
\end{aligned} \tag{17}$$

Fortunately, for maximisation with respect to T and V, the last two difficult terms are merely constant, and we need only to maximise the simpler objective

$$Q(T, V) = \sum_{\nu}\sum_{\tau}\left(\sum_{i}\left(-t_{\nu,i}v_{i,\tau} + \langle s_{\nu,i,\tau}\rangle^{(n)}\log(t_{\nu,i}v_{i,\tau})\right)\right), \tag{18}$$

where we only need the expected value of the sources given by the previous values of the templates and excitations:

$$\langle s_{\nu,i,\tau}\rangle^{(n)} = x_{\nu,\tau}\frac{t_{\nu,i}^{(n)}v_{i,\tau}^{(n)}}{\sum_{i'}t_{\nu,i'}^{(n)}v_{i',\tau}^{(n)}}. \tag{19}$$

Maximisation of the objective Q and substituting $\langle s_{\nu,i,\tau}\rangle^{(n)}$ give the following fixed point equations:

$$\begin{aligned}
\frac{\partial Q}{\partial t_{\nu,i}} &= -\sum_{\tau}v_{i,\tau}^{(n)} + \frac{\sum_{\tau}\langle s_{\nu,i,\tau}\rangle^{(n)}}{t_{\nu,i}}, \\
t_{\nu,i}^{(n+1)} &= \frac{\sum_{\tau}\langle s_{\nu,i,\tau}\rangle^{(n)}}{\sum_{\tau}v_{i,\tau}^{(n)}} = t_{\nu,i}^{(n)}\frac{\sum_{\tau}x_{\nu,\tau}v_{i,\tau}^{(n)}/\sum_{i'}t_{\nu,i'}^{(n)}v_{i',\tau}^{(n)}}{\sum_{\tau}v_{i,\tau}^{(n)}}, \\
\frac{\partial Q}{\partial v_{i,\tau}} &= -\sum_{\nu}t_{\nu,i}^{(n)} + \frac{\sum_{\nu}\langle s_{\nu,i,\tau}\rangle^{(n)}}{v_{i,\tau}}, \\
v_{i,\tau}^{(n+1)} &= \frac{\sum_{\nu}\langle s_{\nu,i,\tau}\rangle^{(n)}}{\sum_{\nu}t_{\nu,i}^{(n)}} = v_{i,\tau}^{(n)}\frac{\sum_{\nu}t_{\nu,i}^{(n)}x_{\nu,\tau}/\sum_{i'}t_{\nu,i'}^{(n)}v_{i',\tau}^{(n)}}{\sum_{\nu}t_{\nu,i}^{(n)}}.
\end{aligned} \tag{20}$$

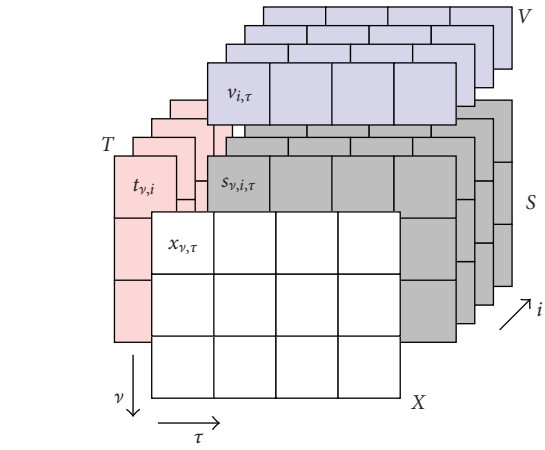
Equation (20) is identical to the multiplicative update rules of [8]. However, our derivation via data augmentation obtains the same result as an EM algorithm. It is interesting to note that in literature, NMF is often described as EM-like; here, we show that it is actually just an EM algorithm. We see that the efficiency of NMF is due to the fact that the $W \times I \times K$ object $\langle S\rangle$ needs not to be explicitly calculated as we only need its marginal statistics (sums across τ or ν).

We note that our model is valid when X is integer valued. See [9] for a detailed discussion about consequences of this issue. Here, we assume that for nonnegative real valued $\widetilde{X}$, we only consider the integer part, that is, we let $\widetilde{X} = X + E$, where E is a noise matrix with entries uniformly drawn in $[0, 1)$. In practice, this is not an obstacle when the entries of X are large.

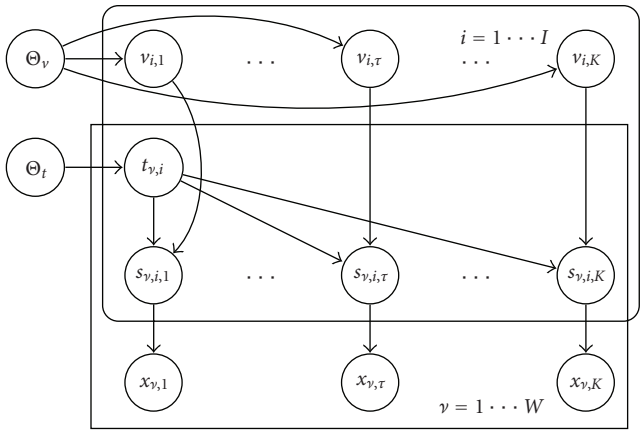
The interpretation of NMF as a maximum likelihood method in a Poisson model is mentioned in the original NMF paper [1] and discussed in more detail by [5, 10]. The equivalence of NMF and probabilistic latent sematic analysis is shown in [11]. Kameoka in [5] focuses on the optimisation and gives an equivalent description using auxiliary function maximisation. In contrast, the auxiliary variables can be viewed as model variables (the sources s) that are analytically integrated out [10]. A general framework is described in [12]. Prior structures are placed on conditionally Gaussian NMF models to enforce sparsity in [13]. However, all of these approaches are based on regularisation, that is, aim at calculating a maximum a posteriori estimate. In contrast, we provided in this article a full Bayesian treatment where the templates and excitations are integrated out.

2.2. Hierarchical Prior Structure. Given the probabilistic interpretation, it is possible to propose various hierarchical prior structures to fit the requirements of an application. Here we will describe a simple choice where we have a conjugate prior as follows:

$$t_{\nu,i} \sim \mathcal{G}\left(t_{\nu,i}; a_{\nu,i}^{t}, \frac{b_{\nu,i}^{t}}{a_{\nu,i}^{t}}\right), \qquad v_{i,\tau} \sim \mathcal{G}\left(v_{i,\tau}; a_{i,\tau}^{v}, \frac{b_{i,\tau}^{v}}{a_{i,\tau}^{v}}\right). \tag{21}$$



(a)



(b)

Figure 1: (a) A schematic description of the NMF model with data augmentation. (b) Graphical model with hyperparameters. Each source element $s_{\nu,i,\tau}$ is Poisson distributed with intensity $t_{\nu,i}v_{i,\tau}$. The observations are given by $x_{\nu,\tau} = \sum_i s_{\nu,i,\tau}$. In matrix notation, we write $X = \sum S_i$. We can analytically integrate out over S. Due to superposition property of Poisson distribution, intensities add up, and we obtain $\langle X \rangle = TV$. Given X, the NMF algorithm is shown to seek the maximum likelihood estimates of the templates T and excitations V. In our Bayesian treatment, we further assume that elements of T and V are Gamma distributed with hyperparameters Θ.

Here, $\mathcal{G}$ denotes the density of a gamma random variable $x \in \mathbb{R}_+$ with shape $a \in \mathbb{R}_+$ and scale $b \in \mathbb{R}_+$ defined by

$$\mathcal{G}(x; a, b) = \exp\left((a-1)\log x - \frac{x}{b} - \log\Gamma(a) - a\log b\right). \tag{22}$$

The primary motivation for choosing a Gamma distribution is computational convenience: Gamma distribution is the conjugate prior to Poisson intensity. The indexing highlights the most general case where there are individual parameters for each element $t_{\nu,i}$ and $v_{i,\tau}$. Typically, we do not allow many free hyperparameters but tie them depending upon the requirements of an application. See Figure 1 for an example. As an example, consider a model where we tie

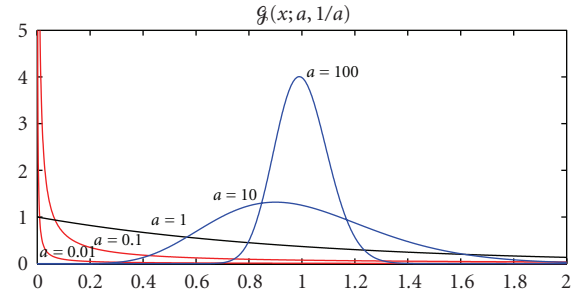


Figure 2: (Left) The family of densities $p(v; a, b = 1) = \mathcal{G}(v; a, b/a)$ with the same mean $\langle v \rangle = b = 1$. Small values of a (for $a < 1$) enforce sparser representations, and large values of $a \approx 100$ tie all values to be close to a nonzero mean (nonsparse representation).

the hyperparameters such as $a^t_{\nu,i} = a^t$, $b^t_{\nu,i} = b^t$, $a^v_{i,\tau} = a^v$, and $b^v_{i,\tau} = b^v$ for $i = 1 \cdots I$, $\nu = 1 \cdots W$, and $\tau = 1 \cdots K$. This model is simple to interpret, where each component of the templates and the excitations is drawn independently from the Gamma family shown in Figure 2. Qualitatively, the shape parameter a controls the *sparsity* of the representation. Remember that $\mathcal{G}(x; a, b/a)$ has the mean b and standard deviation $b/\sqrt{a}$. Hence, for large a, all coefficients will have more or less the same magnitude b, and typical representations will be full. In contrast, for small a, most of the coefficients will be very close to zero, and only very few will be dominating, hence favouring a sparse representation. The scale parameter b is adapted to give the expected magnitude of each component.

To model missing data, that is, when some of the $x_{\nu,\tau}$ are not observed, we define a *mask* matrix $M = \{m_{\nu,\tau}\}$, the same size as X where $m_{\nu,\tau} = 0$, if $x_{\nu,\tau}$ is missing and 1 otherwise (see Appendix A.4 for details). Using the mask variables, the observation model with missing data can be written as

$$p(X \mid S)p(S \mid T, V) = \prod_{\nu,\tau} \left(p(x_{\nu,\tau} \mid s_{\nu,1:I,\tau})\, p(s_{\nu,1:I,\tau} \mid t_{\nu,1:I}, v_{1:I,\tau})\right)^{m_{\nu,\tau}}. \tag{23}$$

The hierarchical model in (21) is more powerful than the basic model of (5), in that it allows a lot of freedom for more realistic modelling. First of all, the hyperparameters can be estimated from examples of a certain class of source to capture the invariant features. Another possibility is Bayesian model selection, where we can compare alternative models in terms of their marginal likelihood. This enables one to estimate the model order, for example, the optimum number of templates to represent a source.

3. Full Bayesian Inference

Below, we describe various interesting problems that can be cast to Bayesian inference problems. In signal analysis and feature extraction with NMF, we may wish to calculate the posterior distribution of templates and excitations, given data and hyperparameters $\Theta \equiv (\Theta^t, \Theta^v)$. Another

(1) Initialise:
$L_t^{(0)} = E_t^{(0)} \sim \mathcal{G}(\cdot; A_t, B_t./A_t) \qquad L_v^{(0)} = E_v^{(0)} \sim \mathcal{G}(\cdot; A_v, B_v./A_v)$
(2) **for** $n = 1 \ldots$ MAXITER **do**
(3) Source sufficient statistics
$\Sigma_t^{(n)} := L_t^{(n-1)} .* (((X .* M)./(L_t^{(n-1)} L_v^{(n-1)}))L_v^{(n-1)^\top})$
$\Sigma_v^{(n)} := L_v^{(n-1)} .* (L_t^{(n-1)^\top}((X .* M)./(L_t^{(n-1)} L_v^{(n-1)})))$
(4) Means
$E_t^{(n)} := \boldsymbol{\alpha}_t^{(n)} .* \boldsymbol{\beta}_t^{(n)} \qquad \boldsymbol{\alpha}_t^{(n)} = A_t + \Sigma_t^{(n)} \qquad \boldsymbol{\beta}_t^{(n)} = 1./(A_t./B_t + ME_v^{(n-1)^\top})$
$E_v^{(n)} := \boldsymbol{\alpha}_v^{(n)} .* \boldsymbol{\beta}_v^{(n)} \qquad \boldsymbol{\alpha}_v^{(n)} = A_v + \Sigma_v^{(n)} \qquad \boldsymbol{\beta}_v^{(n)} = 1./(A_v./B_v + E_t^{(n)^\top} M)$
(5) Optional: Compute Bound (See Appendix, (A.13))
(6) Means of Logs
$L_t^{(n)} = \exp(\Psi(\boldsymbol{\alpha}_t^{(n)})) .* \boldsymbol{\beta}_t^{(n)} \qquad L_v^{(n)} = \exp(\Psi(\boldsymbol{\alpha}_v^{(n)})) .* \boldsymbol{\beta}_v^{(n)}$
(7) Optional: Update Hyperparameters (See Appendix, Section A.5)
(8) **end for**

Algorithm 1: Variational nonnegative matrix factorisation.

important quantity is the marginal likelihood (also known as the *evidence*), where

$$p(X \mid \Theta) = \int dT\, dV \sum_S p(X \mid S)p(S \mid T, V)p(T, V \mid \Theta). \tag{24}$$

The marginal likelihood can be used to estimate the hyperparameters, given examples of a source class

$$\Theta^* = \arg\max_\Theta p(X \mid \Theta) \tag{25}$$

or to compare two given models via *Bayes factors*

$$l(\Theta_1, \Theta_2) = \frac{p(X \mid \Theta_1)}{p(X \mid \Theta_2)}. \tag{26}$$

This latter quantity is particularly useful for comparing different classes of models. Unfortunately, the integrations required cannot be computed in closed form. In the sequel, we will describe the Gibbs sampler and variational Bayes as approximate inference strategies.

3.1. Variational Bayes. We sketch here the Variational Bayes (VB) [3, 14] method to bound the marginal log-likelihood as

$$\begin{aligned}\mathcal{L}_X(\Theta) \equiv \log p(X \mid \Theta) &\geq \sum_S \int d(T, V) q \log \frac{p(X, S, T, V \mid \Theta)}{q} \\ &= \langle \log p(X, S, V, T \mid \Theta) \rangle_q + H[q] \equiv \mathcal{B}_{\mathrm{VB}}[q],\end{aligned} \tag{27}$$

where $q = q(S, T, V)$ is an instrumental distribution, and $H[q]$ is its entropy. The bound is tight for the exact posterior $q(S, T, V) = p(S, T, V \mid X, \Theta)$ but as this distribution is complex, we assume a factorised form for the instrumental distribution by ignoring some of the couplings present in the exact posterior as follows:

$$\begin{aligned}q(S, T, V) &= q(S)q(T)q(V) \\ &= \left(\prod_{\nu,\tau} q(s_{\nu,1:I,\tau})\right)\left(\prod_{\nu,i} q(t_{\nu,i})\right)\left(\prod_{i,\tau} q(v_{i,\tau})\right) \equiv \prod_{\boldsymbol{\alpha}\in\mathcal{C}} q_{\boldsymbol{\alpha}},\end{aligned} \tag{28}$$

where $\boldsymbol{\alpha} \in \mathcal{C} = \{\{S\}, \{T\}, \{V\}\}$ denotes set of disjoint clusters. Hence, we are no longer guaranteed to attain the exact marginal likelihood $\mathcal{L}_X(\Theta)$. Yet, the bound property is preserved, and the strategy of VB is to optimise the bound. Although the best q distribution respecting the factorisation is not available in closed form, it turns out that a local optimum can be attained by the following fixed point iteration:

$$q_{\boldsymbol{\alpha}}^{(n+1)} \propto \exp\left(\langle \log p(X, S, T, V \mid \Theta) \rangle_{q_{\neg\boldsymbol{\alpha}}^{(n)}}\right), \tag{29}$$

where $q_{\neg\boldsymbol{\alpha}} = q/q_{\boldsymbol{\alpha}}$. This iteration monotonically improves the individual factors of the q distribution, that is, $\mathcal{B}[q^{(n)}] \leq \mathcal{B}[q^{(n+1)}]$ for $n = 1, 2, \ldots$ given an initialisation $q^{(0)}$. The order is not important for convergence; one could visit blocks in arbitrary order. However, in general, the attained fixed point depends upon the order of the updates as well as the starting point $q^{(0)}(\cdot)$. We choose the following update order in our derivations:

$$q(S)^{(n+1)} \propto \exp\left(\langle \log p(X, S, T, V \mid \Theta) \rangle_{q(T)^{(n)} q(V)^{(n)}}\right), \tag{30}$$

$$q(T)^{(n+1)} \propto \exp\left(\langle \log p(X, S, T, V \mid \Theta) \rangle_{q(S)^{(n+1)} q(V)^{(n)}}\right), \tag{31}$$

$$q(V)^{(n+1)} \propto \exp\left(\langle \log p(X, S, T, V \mid \Theta) \rangle_{q(S)^{(n+1)} q(T)^{(n+1)}}\right). \tag{32}$$

3.2. Variational Update Equations and Sufficient Statistics. The expectations of $\langle \log p(X, S, T, V \mid \Theta) \rangle$ are functions

of the sufficient statistics of q (see the expression in the Appendix A.2). The update equation for the latent sources (30) leads to the following:

$$\begin{aligned} q(s_{\nu,1:I,\tau}) &\propto \exp\Big(\sum_i (s_{\nu,i,\tau}(\langle \log t_{\nu,i}\rangle + \langle \log v_{i,\tau}\rangle) \\ &\qquad - \log\Gamma(s_{\nu,i,\tau}+1))\Big)\delta\Big(x_{\nu,\tau} - \sum_i s_{\nu,i,\tau}\Big) \\ &\propto \mathcal{M}(s_{\nu,1,\tau},\ldots,s_{\nu,i,\tau},\ldots,s_{\nu,I,\tau}; \\ &\qquad x_{\nu,\tau}, p_{\nu,1,\tau},\ldots,p_{\nu,i,\tau},\ldots,p_{\nu,I,\tau}), \\ p_{\nu,i,\tau} &= \frac{\exp(\langle \log t_{\nu,i}\rangle + \langle \log v_{i,\tau}\rangle)}{\sum_i \exp(\langle \log t_{\nu,i}\rangle + \langle \log v_{i,\tau}\rangle)}, \\ \langle s_{\nu,i,\tau}\rangle &= x_{\nu,\tau} p_{\nu,i,\tau}. \end{aligned} \tag{33}$$

These equations are analogous to the multinomial posterior of EM given in (14); only the computation of cell probabilities is different. The excitation and template distributions and their sufficient statistics follow from the properties of the gamma distribution:

$$\begin{aligned} q(t_{\nu,i}) &\propto \exp\Big(\Big(a^t_{\nu,i} + \sum_\tau \langle s_{\nu,i,\tau}\rangle - 1\Big)\log(t_{\nu,i}) \\ &\qquad - \Big(\frac{a^t_{\nu,i}}{b_{\nu,i}} + \sum_\tau \langle v_{i,\tau}\rangle\Big)t_{\nu,i}\Big) \\ &\propto \mathcal{G}(t_{\nu,i}; \alpha^t_{\nu,i}, \beta^t_{\nu,i}), \\ \alpha^t_{\nu,i} &\equiv a^t_{\nu,i} + \sum_\tau \langle s_{\nu,i,\tau}\rangle, \quad \beta^t_{\nu,i} \equiv \Big(\frac{a^t_{\nu,i}}{b^t_{\nu,i}} + \sum_\tau \langle v_{i,\tau}\rangle\Big)^{-1}, \\ \exp(\langle \log t_{\nu,i}\rangle) &= \exp(\Psi(\alpha^t_{\nu,i}))\beta^t_{\nu,i}, \\ \langle t_{\nu,i}\rangle &= \alpha^t_{\nu,i}\beta^t_{\nu,i}, \\ q(v_{i,\tau}) &\propto \exp\Big(\Big(a^v_{i,\tau} + \sum_\nu \langle s_{\nu,i,\tau}\rangle - 1\Big)\log v_{i,\tau} \\ &\qquad - \Big(\frac{a^v_{i,\tau}}{b^v_{i,\tau}} + \sum_\nu \langle t_{\nu,i}\rangle\Big)v_{i,\tau}\Big) \\ &\propto \mathcal{G}(v_{i,\tau}; \alpha^v_{i,\tau}, \beta^v_{i,\tau}), \\ \alpha^v_{i,\tau} &\equiv a^v_{i,\tau} + \sum_\nu \langle s_{\nu,i,\tau}\rangle, \quad \beta^v_{i,\tau} \equiv \Big(\frac{a^v_{i,\tau}}{b^v_{i,\tau}} + \sum_\nu \langle t_{\nu,i}\rangle\Big)^{-1}, \\ \exp(\langle \log v_{i,\tau}\rangle) &= \exp(\Psi(\alpha^v_{i,\tau}))\beta^v_{i,\tau}, \\ \langle v_{i,\tau}\rangle &= \alpha^v_{i,\tau}\beta^v_{i,\tau}. \end{aligned} \tag{34}$$

3.3. Efficient Implementation. One of the attractive features of NMF is easy and efficient implementation. In this section, we derive that the update equations of Section 3.2 in compact matrix notation are to illustrate that these attractive properties are retained for the full Bayesian treatment. A subtle but key point in the efficiency of the algorithm is that we can avoid explicitly storing and computing the $W \times I \times K$ object $\langle S\rangle$, as we only need the marginal statistics during optimisation. Consider (33). We can write

$$\begin{aligned} \sum_\tau \langle s_{\nu,i,\tau}\rangle &= \sum_\tau x_{\nu,\tau} p_{\nu,i,\tau} \\ &= \exp(\langle \log t_{\nu,i}\rangle) \\ &\qquad \times \sum_\tau \Big(\frac{x_{\nu,\tau}}{(\sum_{i'} \exp(\langle \log t_{\nu,i'}\rangle)\exp(\langle \log v_{i',\tau}\rangle))}\Big) \\ &\qquad \times \exp(\langle \log v_{i,\tau}\rangle), \\ \Sigma_t &= L_t .* ((X./(L_t L_v))L_v^\top). \end{aligned} \tag{35}$$

Here, the denominator has to be nonzero. In the last line, we have represented the expression in compact notation where we define the following matrices:

$$\begin{aligned} E_t &= \{\langle t_{\nu,i}\rangle\}, \qquad L_t = \{\exp(\langle \log t_{\nu,i}\rangle)\}, \\ \Sigma_t &= \Big\{\sum_\tau \langle s_{\nu,i,\tau}\rangle\Big\}, \qquad A_t = \{a^t_{\nu,i}\}, \\ B_t &= \{b^t_{\nu,i}\}, \qquad \boldsymbol{\alpha}_t = \{\alpha^t_{\nu,i}\}, \qquad \boldsymbol{\beta}_t = \{\beta^t_{\nu,i}\}, \\ E_v &= \{\langle v_{i,\tau}\rangle\}, \qquad L_v = \{\exp(\langle \log v_{i,\tau}\rangle)\}, \\ \Sigma_v &= \Big\{\sum_\nu \langle s_{\nu,i,\tau}\rangle\Big\}, \qquad A_v = \{a^v_{i,\tau}\}, \\ B_v &= \{b^v_{i,\tau}\}, \qquad \boldsymbol{\alpha}_v = \{\alpha^v_{i,\tau}\}, \qquad \boldsymbol{\beta}_v = \{\beta^v_{i,\tau}\}. \end{aligned} \tag{36}$$

The matrices subscripted with t are in $\mathbb{R}_+^{W\times I}$ and with v are in $\mathbb{R}_+^{I\times K}$. For notational convenience, we define $.*$ and $./$ as elementwise matrix multiplication and division, respectively, and $\mathbf{1}_W$ as a $W \times 1$ vector of ones. After straightforward substitutions, we obtain the *variational nonnegative matrix factorisation* algorithm, that can compactly be expressed as in panel Algorithm 1.

Similarly, an iterative conditional modes (ICM) algorithm can be derived to compute the maximum a posteriori (MAP) solution (see Appendix A.4):

$$\begin{aligned} V &:= (A_v + V .* (T^\top((M .* X)./(TV)))) \\ &\quad ./(A_v./B_v + T^\top M), \end{aligned} \tag{37}$$

$$\begin{aligned} T &:= (A_t + T .* (((M .* X)./(TV))V^\top)) \\ &\quad ./(A_t./B_t + MV^\top). \end{aligned} \tag{38}$$

Note that when the shape parameters go to zero, that is, $A_t, A_v \to \mathbf{0}$, we obtain the maximum likelihood NMF algorithm.

3.4. Markov Chain Monte Carlo, the Gibbs Sampler. Monte Carlo methods [15, 16] are powerful computational techniques to estimate expectations of form

$$E = \langle f(x)\rangle_{p(x)} \approx \frac{1}{N}\sum_{n=1}^N f(x^{(i)}) = \widetilde{E}_N, \tag{39}$$

(1) Initialize:
$T^{(0)} = \sim \mathcal{G}(\cdot; A_t, B_t) \qquad V^{(0)} \sim \mathcal{G}(\cdot; A_v, B_v)$
(2) **for** $n = 1 \ldots$ MAXITER **do**
(3) Sample Sources
(4) **for** $\tau = 1 \cdots K, \nu = 1 \cdots W$ **do**
(5) $p^{(n)}_{\nu,1:I,\tau} = T^{(n-1)}(\nu, 1:I) .* V^{(n-1)}(1:I,\tau)^{\top} ./ (T^{(n-1)}(\nu, 1:I) V^{(n-1)}(1:I,\tau))$
(6) $S^{(n)}(\nu, 1:I, \tau) \sim \mathcal{M}(s_{\nu,1:I,\tau}; x_{\nu,\tau}, p^{(n)}_{\nu,1:I,\tau})$
(7) **end for**
$\Sigma^{(n)}_t = \sum_{\tau} S^{(n)}_{\nu,i,\tau} \qquad \Sigma^{(n)}_v = \sum_{\nu} S^{(n)}_{\nu,i,\tau}$
(8) Sample Templates
$\boldsymbol{\alpha}^{(n)}_t = A_t + \Sigma^{(n)}_t \qquad \boldsymbol{\beta}^{(n)}_t = 1./(A_t./B_t + \mathbf{1}_W (V^{(n-1)} \mathbf{1}_K)^{\top})$
$T^{(n)} \sim \mathcal{G}(T; \boldsymbol{\alpha}^{(n)}_t, \boldsymbol{\beta}^{(n)}_t)$
(9) Sample Excitations
$\boldsymbol{\alpha}^{(n)}_v = A_v + \Sigma^{(n)}_v \qquad \boldsymbol{\beta}^{(n)}_v = 1./(A_v./B_v + (\mathbf{1}^{\top}_W T^{(n-1)})^{\top} \mathbf{1}^{\top}_K)$
$V^{(n)} \sim \mathcal{G}(V; \boldsymbol{\alpha}^{(n)}_v, \boldsymbol{\beta}^{(n)}_v)$
(10) **end for**

Algorithm 2: Gibbs sampler for nonnegative matrix factorisation.

where $x^{(i)}$ are independent samples drawn from $p(x)$. Under mild conditions on f, the estimate $\widetilde{E}_N$ converges to the true expectation for $N \to \infty$. The difficulty here is obtaining independent samples $\{x^{(i)}\}_{i=1\cdots N}$ from complicated distributions.

The Markov Chain Monte Carlo (MCMC) techniques generate subsequent samples from a Markov chain defined by a *transition kernel* $\mathcal{T}$, that is, one generates $x^{(i+1)}$ conditioned on $x^{(i)}$ as follows:

$$x^{(i+1)} \sim \mathcal{T}(x \mid x^{(i)}). \tag{40}$$

Note that the transition kernel $\mathcal{T}$ is not needed explicitly in practice; all is needed is a procedure to sample a new configuration, given the previous one. Perhaps surprisingly, even though subsequent samples are correlated, provided that $\mathcal{T}$ satisfies certain ergodicity conditions, (39) remains still valid, and estimated expectations converge to their true values when number of samples N goes to infinity [15]. To design a transition kernel $\mathcal{T}$ such that the desired distribution is the stationary distribution, that is, $p(x) = \int dx' \mathcal{T}(x \mid x') p(x')$, many alternative strategies can be employed [16]. One particularly convenient and simple procedure is the Gibbs sampler where one samples each block of variables from *full conditional distributions*. For the NMF model, a possible Gibbs sampler is

$$\begin{aligned} S^{(n+1)} &\sim p(S \mid T^{(n)}, V^{(n)}, X, \Theta), \\ T^{(n+1)} &\sim p(T \mid V^{(n)}, S^{(n+1)}, X, \Theta), \\ V^{(n+1)} &\sim p(V \mid S^{(n+1)}, T^{(n+1)}, X, \Theta). \end{aligned} \tag{41}$$

Note that this procedure implicitly defines a transition kernel $\mathcal{T}(\cdot \mid \cdot)$. It can be shown [15] that the stationary distribution of $\mathcal{T}$ is the exact posterior $p(S, T, V \mid X, \Theta)$. Eventually, the Gibbs sampler converges regardless of the order that the blocks are visited, provided that each block is visited infinitely often in the limit $n \to \infty$. However, the rate of convergence is very difficult to assess as it depends upon the order of the updates as well as the starting configuration $(T^{(0)}, V^{(0)}, S^{(0)})$. It is instructive to contrast above (41) with the variational update of (30)–(32): algorithmically the two approaches are quite similar. The pseudo-code is given in Algorithm 2.

3.4.1. Marginal Likelihood Estimation with Chib's Method. The marginal likelihood can be estimated from the samples generated by the Gibbs sampler using a method proposed by Chib [17]. Suppose we have run the block Gibbs sampler until convergence and have N samples as follows:

$$\{T^{(n)}\}_{n=1:N}, \qquad \{V^{(n)}\}_{n=1:N}, \qquad \{S^{(n)}\}_{n=1:N}. \tag{42}$$

The marginal likelihood is (omitting hyperparameters Θ)

$$p(X) = \frac{p(V, T, S, X)}{p(V, T, S \mid X)}. \tag{43}$$

This equation holds for *all* points (V, T, S). We choose a point in the configuration space; provided that the distribution is unimodal, a good candidate is a configuration near the mode $(\widetilde{T}, \widetilde{V}, \widetilde{S})$. The numerator in (43) is easy to evaluate. The denominator is

$$\begin{aligned} p(\widetilde{V}, \widetilde{T}, \widetilde{S} \mid X) &= p(\widetilde{V} \mid \widetilde{T}, \widetilde{S}, X) p(\widetilde{T} \mid \widetilde{S}, X) p(\widetilde{S} \mid X) \\ &= p(\widetilde{V} \mid \widetilde{T}, \widetilde{S}) p(\widetilde{T} \mid \widetilde{S}) p(\widetilde{S} \mid X). \end{aligned} \tag{44}$$

The first term is full conditional, so it is available for the Gibbs sampler. The third term is

$$\begin{aligned} p(\widetilde{S} \mid X) &= \int dV\, dT\, p(\widetilde{S} \mid V, T, X) p(V, T \mid X) \\ &\approx \frac{1}{N} \sum_{n=1}^{N} p(\widetilde{S} \mid V^{(n)}, T^{(n)}, X). \end{aligned} \tag{45}$$

The second term is trickier

$$p(\tilde{T} \mid \tilde{S}) = \int dV \, p(\tilde{T} \mid V, \tilde{S}) p(V \mid \tilde{S}). \tag{46}$$

The first term here is full conditional. However, the original Gibbs run gives us only samples from $p(V \mid X)$, not $p(V \mid \tilde{S})$. The idea is to run the Gibbs sampler for M further iterations where we sample from $(V_{\tilde{S}}^{(m)}, T_{\tilde{S}}^{(m)}) \sim p(V, T \mid S = \tilde{S})$, that is, with S clamped at $\tilde{S}$. The resulting estimate is

$$p(\tilde{T} \mid \tilde{S}) \approx \frac{1}{M} \sum_{m=1}^{M} p(\tilde{T} \mid V_{\tilde{S}}^{(m)}, \tilde{S}). \tag{47}$$

Chib's method estimates the marginal likelihood as follows:

$$\begin{aligned} \log p(X \mid \Theta) &= \log p(\tilde{V}, \tilde{T}, \tilde{S}, X \mid \Theta) - \log p(\tilde{V}, \tilde{T}, \tilde{S} \mid X, \Theta) \\ &\approx \log p(\tilde{V}, \tilde{T}, \tilde{S}, X \mid \Theta) - \log p(\tilde{V} \mid \tilde{T}, \tilde{S}, \Theta) \\ &\quad - \log \sum_{m=1}^{M} p(\tilde{T} \mid V_{\tilde{S}}^{(m)}, \tilde{S}, \Theta) \\ &\quad - \log \sum_{n=1}^{N} p(\tilde{S} \mid V^{(n)}, T^{(n)}, X, \Theta) + \log(MN). \end{aligned} \tag{48}$$

4. Simulations

Our goal is to illustrate our approach in a model selection context. We first illustrate that the variational approximation to the marginal likelihood is close to the one obtained from the Gibbs sampler via Chib's method. Then, we compare the quality of solutions we obtain via Variational NMF and compare them to the original NMF on a prediction task. Finally, we focus on reconstruction quality in the overcomplete case where the standard NMF is subject to overfitting.

Model Order Determination. To test our approach, we generate synthetic data from the hierarchical model in (21) with $W = 16$, $K = 10$, and the number of sources $I_{\text{true}} = 5$. The inference task is to find the correct number of sources, given X. The hyperparameters of the true model are set to $a_{v,i}^{t} = a^{t} = 10$, $b_{v,i}^{t} = b^{t} = 1$, $a_{i,\tau}^{v} = a^{v} = 1$, and $b_{i,\tau}^{v} = b^{v} = 100$. In the first experiment, the hyperparameters are assumed to be known and in the second are jointly estimated from data, using hyperparameter adaptation. We evaluate the marginal likelihood for models with the number of templates $I = 1 \cdots 10$, with the Gibbs sampler using Chib's method and variational lower bound $\mathcal{B}$ via variational Bayes. We run the Gibbs sampler for MAXITER = 10 000 steps following a burn-in period of 5000 steps; then we clamp the sources S and continue the simulation for a further 10 000 steps to estimate quantities required by Chib's method. We run the variational algorithm until convergence of the bound or 10 000 iterations, whichever occurs first. In Figure 3(a), we show a comparison of the variational estimate with the average of 5 independent runs obtained via Chib's method. We observe, that both methods give consistent results. In Figure 4, we show the lower bound as a function of model order I, where for each I, the bound is optimised independently by jointly optimising hyperparameters a_t, b_t, a_v, and b_v using the equations derived in the appendix. We observe, that the correct model order can be inferred even when the hyperparameters are unknown a priori. This is potentially useful for estimation of model order from real data.

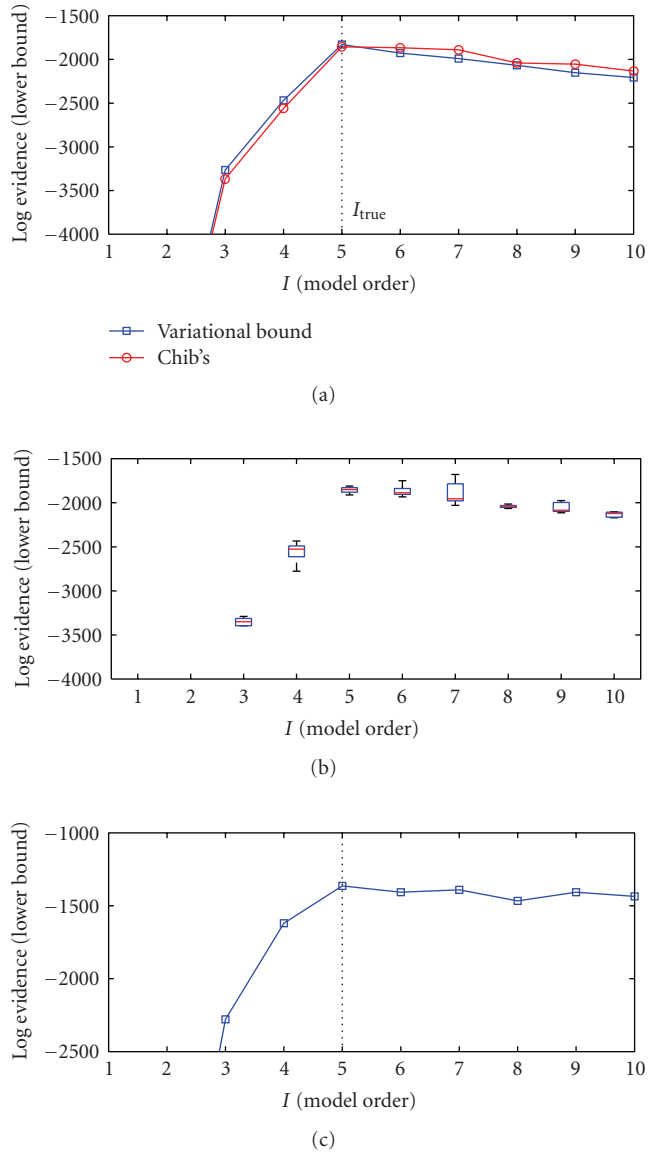


Figure 3: Model selection results. (a) Comparison of model selection by variational bound (squares) and marginal likelihood estimated by Chib's (circles) method. The hyperparameters are assumed to be known. (b) Box-plot of marginal likelihood estimated by Chib's method using 5000, 10 000, and 10 000 iterations for burn-in, free, and clamped sampling. The boxes show the lower quartile, median, and upper quartile values. (c) Model selection by variational bound when hyperparameters are unknown and jointly estimated.

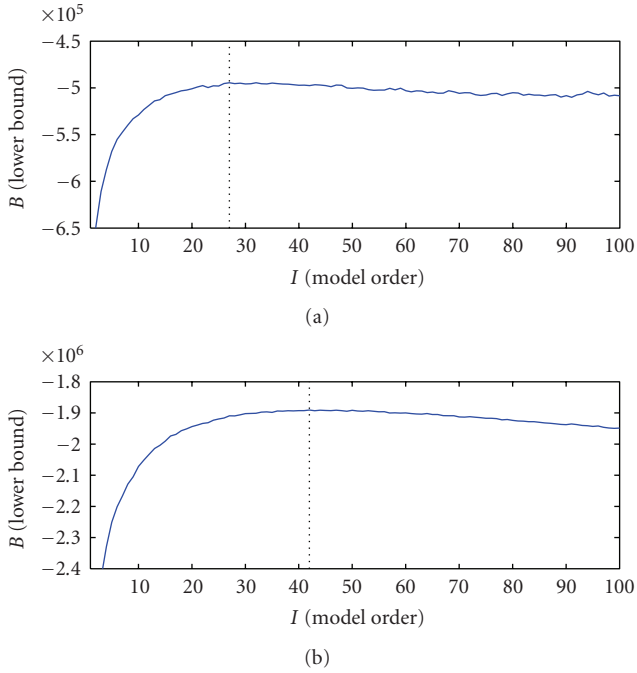


Figure 4: Model selection using variational bound with adapted hyperparameters on face data 16×16 with $I^* = 27$ (a) and 32×32 with $I^* = 42$ (b).

As real data, we use a version of the Olivetti face image database ($K = 400$ images of 64×64 pixels available at http://www.cs.toronto.edu/~roweis/data/olivettifaces.mat). We further downsampled to 16×16 or 32×32 pixels, hence our data matrix X is $16^2 \times 400$ or $32^2 \times 400$. We use a model with tied hyperparameters as $a^t_{\nu,i} = a^t$, $b^t_{\nu,i} = b^t$, $a^v_{i,\tau} = a^v$, and $b^v_{i,\tau} = b^v$, where all hyperparameters are jointly estimated. In Figure 4, bottom, we show results of model order determination for this dataset with joint hyperparameter adaptation. Here, we run the variational algorithm for each model order $I = 1 \cdots 100$ independently and evaluate the lower bound after optimising the hyperparameters. The Gibbs sampler is not found practical and is omitted here. The lower bound behaves as is expected from marginal likelihood, reflecting the tradeoff between too many and too few templates. Higher resolution implies more templates, consistent with our intuition that detail requires more templates for accurate representation.

We also investigate the nature of the representations (see Figure 5). Here, for each independent run, we fix the values of shape parameters to $(a^t, a^v) = [(10, 10), (0.1, 0.1), (10, 0.2), (10, 0.5)]$ and only estimate b^t and b^v. This corresponds to enforcing sparse or nonsparse t and v. Each column shows $I = 36$ templates estimated from the dataset conditioned on hyperparameters. The middle image is the same template image above weighted with the excitations corresponding to the reconstruction (the expected value of the predictive distribution) below. Here, we clearly see the effect of the hyperparameters. In the first condition $(a^t, a^v) = (10, 10)$, the prior does not enforces sparsity to the templates and excitations. Hence, for the representation of a given image, there are many active templates. In the second condition, we try to force both matrices to be sparse with $(a^t, a^v) = (0.1, 0.1)$. Here, the result is not satisfactory as isolated components of the templates are zeroed, giving a representation that looks like one contaminated by "salt-and-pepper" noise. The third condition $((a^t, a^v) = (10, 0.2))$ forces only the excitations to be sparse. Here, we observe that the templates correspond to some average face images. Qualitatively, each image is reconstructed using a superposition of a few of these templates. In the final representation, we enforce sparsity in the templates but not in the excitations. Here, our estimate finds templates that correspond to parts of individual face images (eyebrows, lips, etc.). This solution, intuitively corresponding to a parsimonious representation, also is the best in terms of the marginal likelihood. With proper initialisation, our variational procedure is able to find such solutions.

Prediction. We now compare variational Bayesian NMF with the maximum likelihood NMF on a missing data prediction task.

To illustrate the self regularisation effect, we set up an experiment in which we select a subset of the face data consisting of 50 images. From half of the images, we remove the same patch (Figure 6) and predict the missing pixels. This is a rather small dataset for this task, as we have only 10 images for each of the 5 different persons, and half of these images have missing data at the same spot. We measure the quality of the prediction in terms of signal-to-noise ratio (SNR). The missing values are reconstructed using the mean of the predictive distribution $X_{\text{pred}} \equiv \langle X \rangle_{\mathcal{PO}(X;T^*V^*)} = T^*V^*$, where T^* and V^* are point estimates of the template and excitation matrix. We compare our variational algorithm with the classical NMF. For each algorithm, we test two different versions. The variational algorithms differ in how we estimate T^* and V^*. In the first variational algorithm, we use a crude estimate of T^* and V^* as the mean of the approximating q distribution. In the second condition, after convergence of hyperparameters via VB, we reinitialise T and V randomly and switch to an ICM algorithm (see (38)). This strategy finds a local mode (T^*, V^*) of the exact posterior distribution. In NMF, we test two initialisation strategies: in the first condition, we initialise the templates randomly; in the second, we set them equal to the images in the dataset with random perturbations.

In Figure 6, we show the reconstruction results for a typical run, for a model with $I = 100$ templates. Note that this is an overcomplete model, with twice as many templates as there are images. To characterise the nature of the estimated template and excitation matrices, we use the sparseness criteria [18] of an $m \times n$ matrix X, defined as $\text{Sparseness}(X) = (\sqrt{mn} - (\sum_{i,j}|X_{i,j}|)/(\sum_{i,j}X^2_{i,j})^{1/2})/(\sqrt{mn} - 1)$. This measure is 1 when the matrix X has only a single nonzero entry and 0 when all entries are equal. We see that the variational algorithms are superior in this case in terms of SNR as well as the visual quality of the reconstruction. This is perhaps not surprising, since with

Figure 5: Templates, excitations for a particular example, and the reconstructions obtained for different hyperparameter settings. B is the lower bound for the whole dataset.

maximum likelihood estimation; if the model order is not carefully chosen, generalisation performance is poor: the "noise" in the observed data is fitted but the prediction quality drops on new data. An interesting observation is that highly sparse solutions (either in templates or excitations) do not give the best result; the solution that balances both seems to be the best in this setting. This example illustrates that sparseness in itself may not be necessarily a good criteria to optimise; for model selection, the marginal likelihood should be used as the natural quantity.

On the same face dataset, we compare the prediction error in terms of the SNR for varying model order I. Our goal is to compare the prediction performance of the full Bayesian approach with the ML-NMF for a range of conditions (under-complete, complete, and overcomplete). The results shown in Figure 7 are averages of several runs with hyperparameter adaptation and different hyperparameter tying. In the simulations, the shape parameters are tied always as $a^v_{i,\tau} = a^v$ (and $a^t_{\nu,i} = a^t$). The scale parameters are untied or tied as (b^v_τ, b^t_ν) (across sources) or b^v_i, b^t_i (different for each source) and jointly optimised. Regardless of the hyperparameter tying structure, the results were quite similar. The best SNR values are attained with untied scale parameters for both excitations and templates.

We observe that, due to the implicit self-regularisation in the Bayesian approach, the prediction performance is not very sensitive to the model order and is immune to overfitting. In contrast, the ML-NMF with random initialisation is prone to overfitting, and prediction performance drops with increasing model order. Interestingly, when we initialise the ML-NMF algorithm to true data points with small perturbations, the prediction performance in terms of SNR improves. Note that this strategy would not be possible for data where the pixels were truly missing. However, visual inspection shows that the interpolation can still be "patchy" (see Figure 6).

We observe that hyperparameter adaptation is crucial for obtaining good prediction performance. In our simulations, results for VB without hyperparameter adaptation were occasionally poorer than the ML estimates. Good

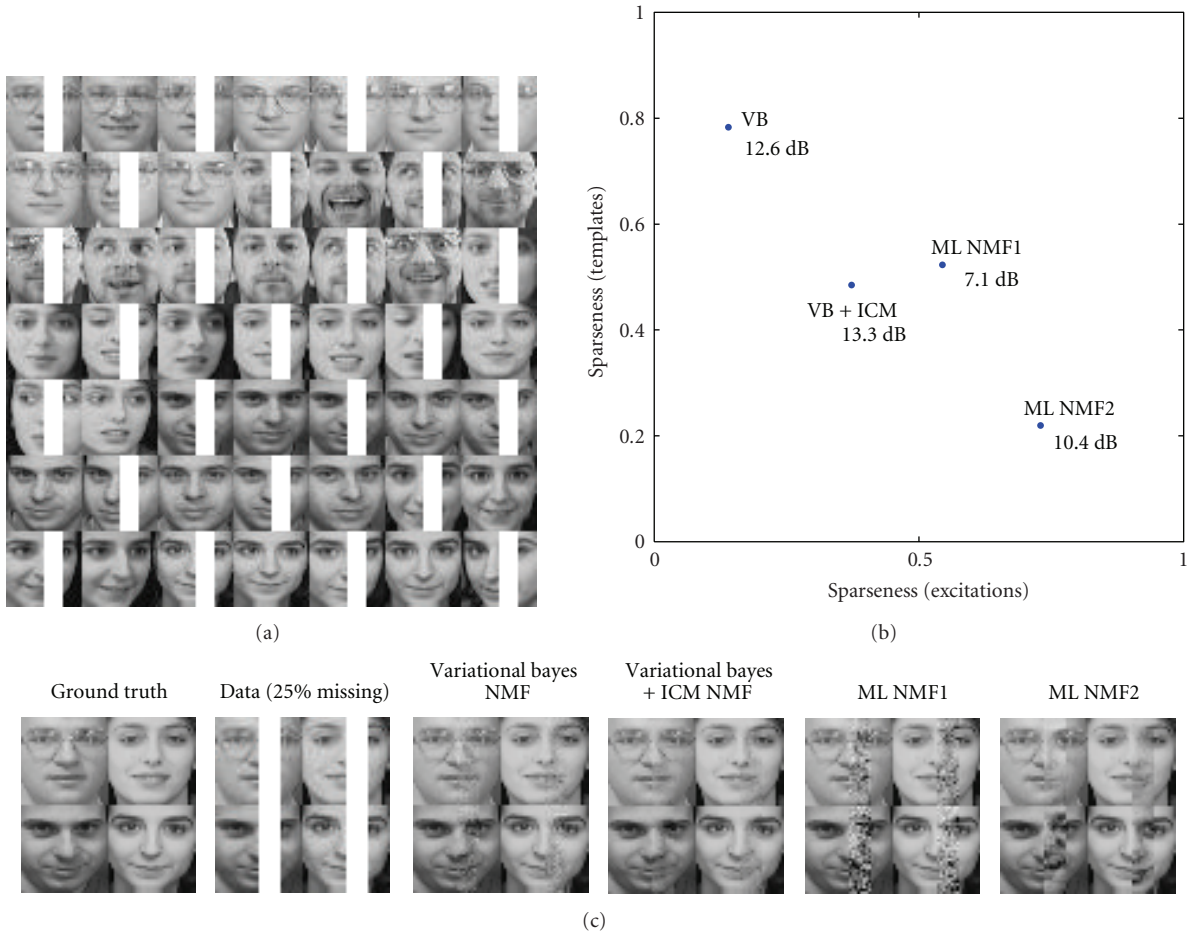


Figure 6: Results of a typical run. (a) Example images from the dataset. (b) Comparison of the reconstruction accuracy of different methods in terms of SNR (in dB), organised according to the sparseness of the solution. (c) (from left to right). The ground truth, data with missing pixels. The reconstructions of VB, VB + ICM, and ML-NMF with two initialisation strategies (1 = random, 2 = to image).

initialisation of the shape hyperparameters seems to be also important. We obtain best results when initialising the shape hyperparameters asymmetrically, for example, $a^v < 1$ and $a^t > 10$ (see 3rd and 4th panels from left in Figure 5). When the shape hyper-parameters are initialised to small $a^v, a^t \ll 1$, the EM seems to get stuck in a local minima more often. Consequently, the prediction results are poorer. We have also carried out tests with more undercomplete representations when the model order is low $I < 10$. For these simulations, while the marginal likelihood was in favour of the VB solutions, we have not observed statistically significant differences between VB and ML in terms of SNR. The SNR improvement of VB over ML was on average about 0.1 dB only.

5. Discussion and Conclusions

In this paper, we have investigated KL-NMF from a statistical perspective. We have shown that KL minimisation formulation the original algorithm can be derived from a probabilistic model where the observations are superposition of I independent Poisson-distributed latent sources. Here, the template and excitation matrices turn out to be latent intensity parameters. The interpretation of NMF as a maximum likelihood method in a Poisson model is mentioned in the original NMF paper [1] and discussed in more detail by [5, 10], and [5] focuses on the optimisation and gives an equivalent description using auxiliary function maximisation. In contrast, [10] illustrates that the auxiliary variables can be viewed as model variables (the sources s) that are analytically integrated out. The relationship between KL divergence and the Poisson distribution is not just a lucky coincidence. There exists a duality between divergence functions and exponential family distributions. If a cost function is a Bregman divergence, there exists a regular exponential family where minimising the cost corresponds to maximum likelihood parameter estimation [19]; also see [12] it in the context of matrix factorisation models.

The novel observation in the current article is the exact characterisation of the approximating distribution $q(S)$ or full conditionals $p(S \mid T, V, X)$ as a product of multinomial distributions, leading to a richer approximation distribution than a naive mean field or single site Gibbs (which would

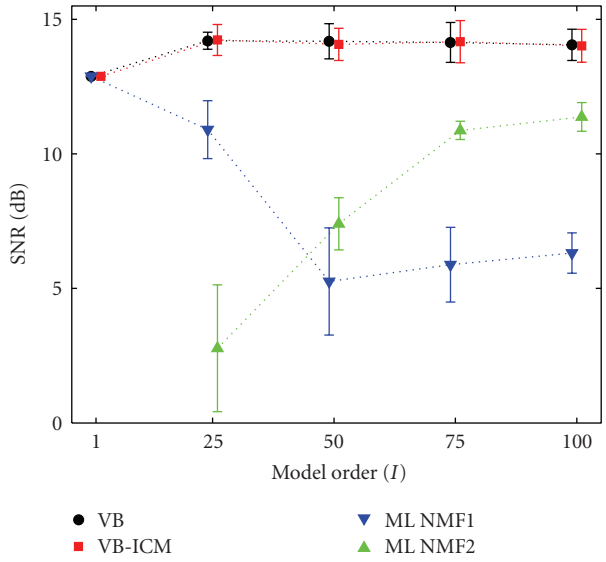


Figure 7: Average SNR results for model orders $I = 1, 25, 50, 75, 100$ covering undercomplete, complete, and overcomplete cases. Comparison of VB, VB + ICM, and ML-NMF with two initialisation strategies (1 = random, 2 = to image).

freeze due to deterministic $p(X \mid S)$). This conjugate form leads to significant simplifications in full Bayesian integration. Apart from the conditionally Gaussian case, NMF with KL objective seems to be unique in this respect. For several other distance metrics $D(\cdot\|\cdot)$, we find that full Bayesian inference not as practical as $p(S \mid T, V, X)$ is not standard.

We have also shown that the standard KL-NMF algorithm with multiplicative update rules is in fact an EM algorithm with data augmentation. Extending upon this observation, we have developed an hierarchical model with conjugate Gamma priors. We have developed a variational Bayes algorithm and a Gibbs sampler for inference in this hierarchical model. We have also developed methods for estimating the marginal likelihood for model selection. This is an additional feature that is lacking in existing NMF approaches with regularisation, where only MAP estimates are obtained, such as [13, 18, 20].

Our simulations suggest that the variational bound seems to be a reasonable approximation to the marginal likelihood and can guide model selection for NMF. The computational requirements are comparable to the ML-NMF. A potentially time-consuming step in the implementation of the variational algorithm is the evaluation of the Ψ function but this step can also be replaced by a simple piecewise polynomial approximation since $\exp(\Psi(x)) \approx x - 0.5$ for $x > 5$.

We first compare the variational inference with a Gibbs sampler. In our simulations, we observe that both algorithms give qualitatively very similar results, both for inference of templates and excitations as well as model order selection. We find the variational approach somewhat more practical as it can be expressed as simple matrix operations, where both the fixed point equations as well as the bound can be compactly and efficiently implemented using matrix computation software. In contrast, our Gibbs sampler is computationally more demanding, and the calculation of marginal likelihood is somewhat more tricky. With our implementation of both algorithms, the variational method is faster by a factor of around 13. Reference implementations of both algorithms in Matlab are available from the following url: http://www.cmpe.boun.edu.tr/~cemgil/bnmf/.

In terms of computational requirements, the variational procedure has several advantages. First, we circumvent sampling from multinomial variables, which is the main computational bottleneck with the Gibbs sampler. Whilst efficient algorithms are developed for multinomial sampling [21], the procedure is time consuming when the number of latent sources I is large. In contrast, the variational method estimates the expected sufficient statistics directly by elementary matrix operations. Another advantage is hyperparameter estimation. In principle, it is possible to maximise the marginal likelihood via a Monte Carlo EM procedure [22, 23]; yet this potentially requires many more iterations. In contrast, the evaluation of the derivatives of the lower bound is straightforward and can be implemented without much additional computational cost.

The efficiency of the Gibbs sampler could be improved by working out the distribution of the sufficient statistics of sources directly (namely, quantities $\sum_\tau s_{\nu,i,\tau}$ or $\sum_\nu s_{\nu,i,\tau}$) to circumvent multinomial sampling. Unfortunately, for the sum of binomial random variables with different cell probability parameters, the sum does not have a simple form but various approximations are possible [24].

Inference based on VB is easy to implement but at the end of the day, the fixed point iteration is just a gradient-based lower bound optimisation procedure, and second order Newton methods can provide more efficient alternatives. For NMF models, there exist many conditional independence relations, hence the Hessian matrix has a special block structure [12]. It is certainly interesting to develop efficient inference methods that make use of the special block structure of the Hessian matrix. However, as our primary goal was a practical full Bayesian treatment, we have not investigated this path yet. Another approach in this direction is using alternative deterministic integration techniques such as expectation propagation (EP) [25]. Those techniques work directly on an approximation of the true marginal likelihood rather than a bound. A related approach known as expectation consistent (EC) inference is used with success in related source separation problems [26].

From a modelling perspective, our hierarchical model provides some attractive properties. It is easy to incorporate prior knowledge about individual latent sources via hyperparameters, and one can easily capture variability in the templates and excitations that is potentially useful for developing robust techniques. The prior structure here is qualitatively similar to an entropic prior [20, 27], and we find qualitatively similar representations to the ones found by NMF reported earlier by [1, 18]. However, none of the above mentioned methods provide an estimate of the marginal likelihood, which is useful for model selection. Our generative model formulation can be extended in various ways to suit the specific needs of particular applications. For

example, one can enforce more structured prior models such as chains or fields [10]. As a second possibility, the Poisson observation model can be replaced with other models such as clipped Gaussian, Gamma, or Gaussians which lead to alternative source separation algorithms. For example, the case of Gaussian sources where the excitations and templates correspond to the variances is discussed in [28].

Our main contribution here is the development of a principled and practical way to estimate both the optimal sparsity criteria and model order, in terms of marginal likelihood. By maximising the bound on marginal likelihood, we have a method where all the hyperparameters can be estimated from data, and the appropriate sparseness criteria is found automatically. We believe that our approach provides a practical improvement to the highly popular KL-NMF algorithm without incurring much additional computational cost.

Appendix

A. Standard Distributions in Exponential Form, Their Sufficient Statistics and Entropies

(i) Gamma

$$\mathcal{G}(\lambda; a, b) \equiv \exp\Big(+ (a-1)\log\lambda - \frac{1}{b}\lambda - \log\Gamma(a) - a\log b\Big),$$

$$\langle\lambda\rangle_{\mathcal{G}} = ab\,\langle\log\lambda\rangle_{\mathcal{G}} = \Psi(a) + \log(b),$$

$$H_{\mathcal{G}}[\lambda] \equiv -\langle\log\mathcal{G}\rangle_{\mathcal{G}} = -(a-1)\Psi(a) + \log b + a + \log\Gamma(a). \tag{A.1}$$

Here, Ψ denotes the digamma function defined as $\Psi(a) \equiv d\log\Gamma(a)/da$.

(ii) Poisson

$$\mathcal{PO}(s;\lambda) = \exp(s\log\lambda - \lambda - \log\Gamma(s+1)),$$
$$\langle s\rangle_{\mathcal{PO}} = \lambda. \tag{A.2}$$

(iii) Multinomial

$$\mathcal{M}(\mathbf{s}; x, \mathbf{p}) = \delta\Big(x - \sum_i s_i\Big)\exp\Big(\log\Gamma(x+1) + \sum_{i=1}^{I}(s_i\log p_i - \log\Gamma(s_i+1))\Big),$$

$$\langle s_i\rangle_{\mathcal{M}} = x p_i. \tag{A.3}$$

Here, $\mathbf{s} = \{s_1, s_2, \ldots, s_I\}$, $\mathbf{p} = \{p_1, p_2, \ldots, p_I\}$, and $p_1 + p_2 + \cdots + p_I = 1$. Here, p_i, $i = 1 \cdots I$ are the cell probabilities, and x is the index parameter where $s_1 + s_2 + \cdots + s_I = x$. The entropy is given as follows:

$$H_{\mathcal{M}}[s_{\nu,1:I,\tau}] = -\log\Gamma(x_{\nu,\tau}+1) - \sum_{i=1}^{I}\langle s_{\nu,i,\tau}\rangle\log p_{\nu,i,\tau} + \sum_{i=1}^{I}\langle\log\Gamma(s_{\nu,i,\tau}+1)\rangle - \Big\langle\log\delta\Big(x_{\nu,\tau} - \sum_i s_{\nu,i,\tau}\Big)\Big\rangle. \tag{A.4}$$

A closed form expression for the entropy is not known due to $\langle\log\Gamma(s+1)\rangle$ terms but asymptotic expansions exist [29, 30]. Computationally efficient sampling from a multinomial distribution is not trivial; see [21] for a comparison of various methods and detailed discussion of tradeoffs.

A.1. Summary of the Generative Model. We have the following. Indices:

$i = 1 \cdots I$, source index;

$\nu = 1 \cdots W$, Row (frequency bin) index;

$\tau = 1 \cdots K$, Column (time frame) index;

$t_{\nu,i}$: template variable at νth row of the ith source

$$t_{\nu,i} \sim \mathcal{G}\Big(t_{\nu,i}; a^t_{\nu,i}, \frac{b^t_{\nu,i}}{a^t_{\nu,i}}\Big); \tag{A.5}$$

$v_{i,\tau}$: excitation variable of the ith source at τth column

$$v_{i,\tau} \sim \mathcal{G}\Big(v_{i,\tau}; a^v_{i,\tau}, \frac{b^v_{i,\tau}}{a^v_{i,\tau}}\Big); \tag{A.6}$$

$s_{\nu,i,\tau}$: source variable of ith source at νth row (frequency bin) and τth column (time frame)

$$s_{\nu,i,\tau} \sim \mathcal{PO}(s_{\nu,i,\tau}; t_{\nu,i}v_{i,\tau}); \tag{A.7}$$

$x_{\nu,\tau}$: observation at νth row (frequency bin) and τth column (time frame)

$$x_{\nu,\tau} \sim \sum_i s_{\nu,i,\tau}. \tag{A.8}$$

A.2. Expression of the Full Joint Distribution. Here, $\phi \equiv p(X, S, T, V \mid \Theta) = p(X \mid S)p(S \mid T, V)p(V \mid \Theta^v)p(V \mid \Theta^t)$,

$$\begin{aligned}\log\phi = {} & \sum_\nu\sum_i\sum_\tau(-t_{\nu,i}v_{i,\tau} + s_{\nu,i,\tau}\log(t_{\nu,i}v_{i,\tau}) - \log\Gamma(s_{\nu,i,\tau}+1)) \\ & + \sum_\nu\sum_\tau\log\delta\Big(x_{\nu,\tau} - \sum_i s_{\nu,i,\tau}\Big) \\ & + \sum_\nu\sum_i + (a^t_{\nu,i}-1)\log t_{\nu,i} - \frac{a^t_{\nu,i}}{b^t_{\nu,i}}t_{\nu,i} - \log\Gamma(a^t_{\nu,i}) \\ & - a^t_{\nu,i}\log\Big(\frac{b^t_{\nu,i}}{a^t_{\nu,i}}\Big) \\ & + \sum_\tau\sum_i + (a^v_{i,\tau}-1)\log v_{i,\tau} - \frac{a^v_{i,\tau}}{b^v_{i,\tau}}v_{i,\tau} - \log\Gamma(a^v_{i,\tau}) \\ & - a^v_{i,\tau}\log\Big(\frac{b^v_{i,\tau}}{a^v_{i,\tau}}\Big).\end{aligned} \tag{A.9}$$

A.3. The Variational Bound. The variational bound in (27) can be written as

$$\mathcal{L}_X(\Theta) \equiv \log p(X \mid \Theta) \geq \langle\log\phi\rangle_q + H[q] = \mathcal{B}_{\mathrm{VB}}, \tag{A.10}$$

where the energy term is given by the expectation of the expression in Appendix A.2, and $H[q]$ denotes the entropy of the variational approximation distribution q where the individual entropies are defined in Appendix A:

$$\begin{aligned} H[q] &= -\langle \log q \rangle \\ &= \sum_{\nu} \sum_{\tau} H_{\mathcal{M}}[s_{\nu,1:I,\tau}] + \sum_{\nu} \sum_{i} H_{\mathcal{G}}[t_{\nu,i}] + \sum_{i} \sum_{\tau} H_{\mathcal{G}}[v_{i,\tau}]. \end{aligned} \tag{A.11}$$

One potential problem is that this expression requires the entropy of a multinomial distribution for which there is no known simple expression. This is due to terms of form $\langle \log \Gamma(s+1) \rangle$ where only asymptotic expansions are known. Fortunately, the difficult terms in the energy term can be canceled by the corresponding terms in the entropy term, and one obtains the following expression that only depends on known sufficient statistics:

$$\begin{aligned} \mathcal{B} = & -\sum_{\nu} \sum_{\tau} \sum_{i} \langle t_{\nu,i} \rangle \langle v_{i,\tau} \rangle \\ & + \sum_{\nu} \sum_{i} \langle \log t_{\nu,i} \rangle \left(a^{t}_{\nu,i} - 1 + \sum_{\tau} \langle s_{\nu,i,\tau} \rangle \right) \\ & + \sum_{\tau} \sum_{i} \langle \log v_{i,\tau} \rangle \left(a^{v}_{i,\tau} - 1 + \sum_{\nu} \langle s_{\nu,i,\tau} \rangle \right) \\ & + \sum_{\nu} \sum_{i} - \frac{a^{t}_{\nu,i}}{b_{\nu,i}} \langle t_{\nu,i} \rangle - \log \Gamma(a^{t}_{\nu,i}) - a^{t}_{\nu,i} \log \left(\frac{b_{\nu,i}}{a^{t}_{\nu,i}} \right) \\ & + \sum_{\tau} \sum_{i} - \frac{a^{v}_{i,\tau}}{b^{v}_{i,\tau}} \langle v_{i,\tau} \rangle - \log \Gamma(a^{v}_{i,\tau}) - a^{v}_{i,\tau} \log \left(\frac{b^{v}_{i,\tau}}{a^{v}_{i,\tau}} \right) \\ & + \sum_{\nu} \sum_{\tau} \left(- \log \Gamma(x_{\nu,\tau} + 1) - \sum_{i=1}^{I} \langle s_{\nu,i,\tau} \rangle \log p_{\nu,i,\tau} \right) \\ & + \sum_{\nu} \sum_{i} (-(\alpha^{t}_{\nu,i} - 1) \Psi(\alpha^{t}_{\nu,i}) + \log \beta^{t}_{\nu,i} + \alpha^{t}_{\nu,i} + \log \Gamma(\alpha^{t}_{\nu,i})) \\ & + \sum_{i} \sum_{\tau} (-(\alpha^{v}_{i,\tau} - 1) \Psi(\alpha^{v}_{i,\tau}) + \log \beta^{v}_{i,\tau} + \alpha^{v}_{i,\tau} + \log \Gamma(\alpha^{v}_{i,\tau})). \end{aligned} \tag{A.12}$$

After some careful manipulations, the following expression is obtained where $\log L$ denotes here elementwise logarithm of matrix L:

$$\begin{aligned} \mathcal{B} = & \sum_{\nu} \sum_{\tau} (- E_t E_v - \log \Gamma(X+1)) \\ & + \sum_{\nu} \sum_{\tau} -X .* (((L_t .* \log(L_t)) L_v + L_t (L_v .* \log(L_v))) \\ & \qquad ./ (L_t L_v) - \log(L_t L_v)) \\ & + \sum_{\nu} \sum_{i} -(A_t ./ B_t) .* E_t - \log \Gamma(A_t) + A_t .* \log(A_t ./ B_t) \\ & + \sum_{\nu} \sum_{i} \boldsymbol{\alpha}_t .* (\log \boldsymbol{\beta}_t + 1) + \log \Gamma(\boldsymbol{\alpha}_t) \\ & + \sum_{i} \sum_{\tau} - (A_v ./ B_v) .* E_v - \log \Gamma(A_v) + A_v .* \log(A_v ./ B_v) \\ & + \sum_{i} \sum_{\tau} \boldsymbol{\alpha}_v .* (\log \boldsymbol{\beta}_v + 1) + \log \Gamma(\boldsymbol{\alpha}_v). \end{aligned} \tag{A.13}$$

A.4. Handling Missing Data and MAP Estimation. When there is missing data, that is, when some of the $x_{\nu,\tau}$ are not observed, computation is still straightforward in our framework and can be accomplished by a simple modification to the original algorithm. We first define a *mask* matrix $\mathbf{M} = \{m_{\nu,\tau}\}$, same size as X, where

$$m_{\nu,\tau} = \begin{cases} 0, & x_{\nu,\tau} \text{ is missing,} \\ 1, & \text{otherwise.} \end{cases} \tag{A.14}$$

Using the mask variables, the observation model with missing data can be written as follows:

$$\begin{aligned} & p(X \mid S) p(S \mid T, V) \\ & \quad = \prod_{\nu,\tau} \left(p(x_{\nu,\tau} \mid s_{\nu,1:I,\tau}) p(s_{\nu,1:I,\tau} \mid t_{\nu,1:I}, v_{1:I,\tau}) \right)^{m_{\nu,\tau}}. \end{aligned} \tag{A.15}$$

The prior is not affected. Hence, we merely replace the first two lines of the expression for the full joint distribution (given in the Appendix A.2) as follows:

$$\begin{aligned} \log \phi = & \sum_{\nu} \sum_{\tau} m_{\nu,\tau} \sum_{i} (- t_{\nu,i} v_{i,\tau} + s_{\nu,i,\tau} \log(t_{\nu,i} v_{i,\tau}) \\ & \qquad - \log \Gamma(s_{\nu,i,\tau} + 1)) \\ & + \sum_{\nu} \sum_{\tau} m_{\nu,\tau} \log \delta \left(x_{\nu,\tau} - \sum_{i} s_{\nu,i,\tau} \right) + \cdots. \end{aligned} \tag{A.16}$$

Consequently, it is easy to see that

$$\begin{aligned} q(v_{i,\tau}) & \propto \exp\left(\left(a^{v}_{i,\tau} + \sum_{\nu} m_{\nu,\tau} \langle s_{\nu,i,\tau} \rangle - 1 \right) \log v_{i,\tau} \right. \\ & \qquad \left. - \left(\frac{a^{v}_{i,\tau}}{b^{v}_{i,\tau}} + \sum_{\nu} m_{\nu,\tau} \langle t_{\nu,i} \rangle \right) v_{i,\tau} \right) \\ & \propto \mathcal{G}(v_{i,\tau}; \alpha^{v}_{i,\tau}, \beta^{v}_{i,\tau}), \\ \alpha^{v}_{i,\tau} & \equiv a^{v}_{i,\tau} + \sum_{\nu} m_{\nu,\tau} \langle s_{\nu,i,\tau} \rangle, \\ \beta^{v}_{i,\tau} & \equiv \left(\frac{a^{v}_{i,\tau}}{b^{v}_{i,\tau}} + \sum_{\nu} m_{\nu,\tau} \langle t_{\nu,i} \rangle \right)^{-1}. \end{aligned} \tag{A.17}$$

By a derivation analogous to one detailed in Section 3.3, we see that the excitation update equations in Algorithm 1, line 4, can be written using matrix notation as follows:

$$\begin{aligned}\Sigma_v &= L_v .* \left(L_t^\top \left((\mathbf{M} .* X)./(L_t L_v)\right)\right)\\ \boldsymbol{\alpha}_v &= A_v + \Sigma_v, \qquad \boldsymbol{\beta}_v = 1./(A_v./B_v + E_t^\top \mathbf{M}).\end{aligned} \tag{A.18}$$

The update rules for the templates are similar. Note that when there is no missing data, we have $\mathbf{M} = \mathbf{1}_W \mathbf{1}_K^\top$ which gives the original algorithm. The bound in (A.13) can also be easily modified for handling missing data. We merely replace $\mathbf{X} \leftarrow \mathbf{M} .* \mathbf{X}$ and the first term $E_t E_v \leftarrow \mathbf{M} .* E_t E_v$.

We conclude this subsection by noting that the standard NMF update equations, given in (20), can be also rewritten to handle missing data as follows:

$$\begin{aligned}V^{(n+1)} &= V^{(n)} .* \left(T^\top \left((\mathbf{M} .* X)./(TV)\right)\right)./(T^\top \mathbf{M}),\\ T^{(n+1)} &= T^{(n)} .* \left(((\mathbf{M} .* X)./(TV))V^\top\right)./(\mathbf{M}V^\top).\end{aligned} \tag{A.19}$$

Here, the denominator has to be nonzero. Similarly, an iterative conditional modes (ICM) algorithm can be derived to compute the maximum a posteriori (MAP) solution as follows:

$$\begin{aligned}V^{(n+1)} &= \left(A_v + V^{(n)} .* \left(T^\top \left((\mathbf{M} .* X)./(TV)\right)\right)\right)\\ &\quad ./(A_v./B_v + T^\top \mathbf{M}),\\ T^{(n+1)} &= \left(A_t + T^{(n)} .* \left(((\mathbf{M} .* X)./(TV))V^\top\right)\right)\\ &\quad ./(A_t./B_t + \mathbf{M}V^\top).\end{aligned} \tag{A.20}$$

Note that when the shape parameters go to zero, that is, $A_t, A_v \to \mathbf{0}$, we obtain the maximum likelihood NMF algorithm.

A.5. Hyperparameter Optimisation. The hyperparameters $\Theta = (\Theta^t, \Theta^v)$ can be estimated by maximising the bound in 13. Below, we will derive the results for the excitations; the results for templates are similar. The solution for shape parameters involves finding the zero of a function $f(a) - c$, where

$$\begin{aligned}f(a) &= \log a - \Psi(a) + 1,\\ a^* &= f^{-1}(c).\end{aligned} \tag{A.21}$$

The solution can be found by Newton's method by iteration of the following fixed point equation:

$$\begin{aligned}a^{(n+1)} &= a^{(n)} - \frac{f(a^{(n)}) - c}{f'(a^{(n)})}\\ &= a^{(n)} - \frac{\log(a^{(n)}) - \Psi(a^{(n)}) + 1 - c}{1/a^{(n)} - \Psi'(a^{(n)})} = a^{(n)} - \Delta^{(n)}.\end{aligned} \tag{A.22}$$

It is well known that Newton iterations can diverge if started away from the root. Occasionally, we observe that a can become negative. If this is the case, we set $\Delta^{(n)} \leftarrow \Delta^{(n)}/2$, and try again. The digamma Ψ function and its derivative Ψ' are available in numeric computation libraries (e.g., in Matlab as `psi(0,a)` and `psi(1,a)`, resp.).

The derivation of the hyperparameter update equations is straightforward:

$$\begin{aligned}\frac{\partial \mathcal{B}}{\partial a_{i,\tau}^v} &= \langle \log v_{i,\tau} \rangle - \frac{1}{b_{i,\tau}^v} \langle v_{i,\tau} \rangle - \Psi(a_{i,\tau}^v)\\ &\quad - \log b_{i,\tau}^v + \log a_{i,\tau}^v + 1 = 0,\\ c_{i,\tau} &= \log a_{i,\tau}^v - \Psi(a_{i,\tau}^v) + 1,\\ c_{i,\tau} &\equiv \frac{\langle v_{i,\tau} \rangle}{b_{i,\tau}^v} - \left(\langle \log v_{i,\tau} \rangle - \log b_{i,\tau}^v\right),\\ \frac{\partial \mathcal{B}}{\partial b_{i,\tau}^v} &= \frac{a_{i,\tau}^v}{(b_{i,\tau}^v)^2} \langle v_{i,\tau} \rangle - a_{i,\tau}^v \frac{1}{b_{i,\tau}^v} = 0,\\ b_{i,\tau}^v &= \langle v_{i,\tau} \rangle.\end{aligned} \tag{A.23}$$

Tying parameters across τ as $a_i^v = a_{i,\tau}^v$ and $b_i^v = b_{i,\tau}^v$ yields

$$\begin{aligned}\frac{\partial \mathcal{B}}{\partial a_i^v} &= \sum_\tau \langle \log v_{i,\tau} \rangle - \sum_\tau \frac{1}{b_{i,\tau}^v} \langle v_{i,\tau} \rangle - K\Psi(a_i^v)\\ &\quad - \sum_\tau \log b_{i,\tau}^v + K = 0,\\ c_i &= \log a_i^v - \Psi(a_i^v) + 1,\\ c_i &= \frac{1}{K} \sum_\tau \left(\frac{\langle v_{i,\tau} \rangle}{b_{i,\tau}^v} - \left(\langle \log v_{i,\tau} \rangle - \log b_{i,\tau}^v\right)\right),\\ \frac{\partial \mathcal{B}}{\partial b_i^v} &= \sum_\tau \frac{a_{i,\tau}^v}{(b_i^v)^2} \langle v_{i,\tau} \rangle - \frac{1}{b_i^v} \sum_\tau a_{i,\tau}^v,\\ b_i^v &= \frac{\sum_\tau a_{i,\tau}^v \langle v_{i,\tau} \rangle}{\sum_\tau a_{i,\tau}^v}.\end{aligned} \tag{A.24}$$

Tying parameters across τ and i, $a^v = a_{i,\tau}^v$, and $b^v = b_{i,\tau}^v$ yields

$$\begin{aligned}c &= \log a^v - \Psi(a^v) + 1,\\ c &= \frac{1}{KI} \sum_\tau \sum_i \left(\frac{\langle v_{i,\tau} \rangle}{b_{i,\tau}^v} - \left(\langle \log v_{i,\tau} \rangle - \log b_{i,\tau}^v\right)\right),\\ \frac{\partial \mathcal{B}}{\partial b^v} &= \sum_i \sum_\tau \frac{a_{i,\tau}^v}{(b^v)^2} \langle v_{i,\tau} \rangle - \frac{1}{b_i^v} \sum_i \sum_\tau a_{i,\tau}^v,\\ b^v &= \frac{\sum_i \sum_\tau a_{i,\tau}^v \langle v_{i,\tau} \rangle}{\sum_i \sum_\tau a_{i,\tau}^v}.\end{aligned} \tag{A.25}$$

The derivation of the template parameters is exactly analogous. We can express the update equations once again in compact matrix notation as follows:

$$Z \longleftarrow E_v^{(n)}./B_v^{(n)} - \log(L_v^{(n)}./B_v^{(n)}),$$

$$C \longleftarrow \begin{cases} Z, & \text{Not tied,} \\ \dfrac{(Z\mathbf{1}_K)}{K}, & \text{Tie columns (over } \tau\text{),} \\ \dfrac{(\mathbf{1}_I^\top Z)}{I}, & \text{Tie rows (over } i\text{),} \\ \dfrac{(\mathbf{1}_I^\top Z\mathbf{1}_K)}{(KI)}, & \text{Tie all (over } \tau \text{ and } i\text{),} \end{cases}$$

$$A_v^{(n+1)} \longleftarrow \texttt{SolveByNewton}\ (A_v^{(n)}, C),$$

$$B_v^{(n+1)} \longleftarrow \begin{cases} E_v^{(n)}, & \text{Not tied,} \\ (((A_v^{(n)} .* E_v^{(n)})\mathbf{1}_K) & \\ \quad ./(A_v^{(n)}\mathbf{1}_K))\mathbf{1}_K^\top, & \text{Tie columns (over } \tau\text{),} \\ \mathbf{1}_I((\mathbf{1}_I^\top(A_v^{(n)} .* E_v^{(n)})) & \\ \quad ./(\mathbf{1}_I^\top A_v^{(n)})), & \text{Tie rows (over } i\text{),} \\ \mathbf{1}_I((\mathbf{1}_I^\top(A_v^{(n)} .* E_v^{(n)})\mathbf{1}_K) & \\ \quad ./(\mathbf{1}_I^\top A_v^{(n)}\mathbf{1}_K))\mathbf{1}_K^\top, & \text{Tie all (over } \tau \text{ and } i\text{).} \end{cases} \tag{A.26}$$

Here, we assume $\texttt{SolveByNewton}(A_0, C)$ is a matrix-valued function that finds root $C_{i,j} = f(A_{i,j})$ for each element of A, starting from the initial matrix A_0. If C is a scalar or vector, it is repeated over the missing index to implement parameter tying. For example, if C is a $I \times 1$ vector and A_0 is $I \times K$, we assume $C_i = c_{i,\tau}$ for all $\tau = 1 \cdots K$, and the output is the same size as A_0. This is only a notational convenience; an actual implementation can be achieved more efficiently. Again, the implementation of the template parameters is exactly analogous; merely replace above the subscripts as $v \leftarrow t$, $(i, \tau) \leftarrow (v, i)$ and $(I, K) \leftarrow (W, I)$.

Acknowledgments

The author would like to thank Nick Whiteley, Tuomas Virtanen, and Paul Peeling for fruitful discussion and for their comments on earlier drafts of this paper. This research is funded by the Engineering and Physical Sciences Research Council (EPSRC) under the grant EP/D03261X/1 and by the Turkish Science, Technology Research Council grant TUBITAK 107E021 and Boğaziçi University Research Fund BAP 09A105P. The research is carried out while the author was with the Signal Processing and Comms. Lab, Department of Engineering, University of Cambridge, UK and Department of Computer Engineering, Boğaziçi University, Turkey.

References

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects with nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[2] P. Paatero and U. Tapper, "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.

[4] T. Virtanen, *Sound source separation in monaural music signals*, Ph.D. thesis, Tampere University of Technology, Tampere, Finland, November 2006.

[5] H. Kameoka, *Statistical approach to multipitch analysis*, Ph.D. thesis, University of Tokyo, Tokyo, Japan, 2007.

[6] A. Cichocki, M. Mørup, P. Smaragdis, W. Wang, and R. Zdunek, "Advances in nonnegative matrix and tensor factorization," *Computational Intelligence and Neuroscience*, vol. 2008, Article ID 852187, 3 pages, 2008.

[7] J. F. C. Kingman, *Poisson Processes*, Oxford Science, Oxford, UK, 1993.

[8] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS '00)*, vol. 13, pp. 556–562, Denver, Colo, USA, October-November 2000.

[9] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "On the relation between divergence-based minimization and maximum-likelihood estimation for the i-divergence," Personal Communication, 2008.

[10] T. Virtanen, A. T. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 1825–1828, Las Vegas, Nev, USA, March-April 2008.

[11] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implication," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 601–602, Salvador, Brazil, August 2005.

[12] A. P. Singh and G. J. Gordon, "A unified view of matrix factorization models," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '08)*, vol. 5212 of *Lecture Notes in Computer Science*, pp. 358–373, Springer, Antwerp, Belgium, September 2008.

[13] M. N. Schmidt and H. Laurberg, "Nonnegative matrix factorization with Gaussian process priors," *Computational Intelligence and Neuroscience*, vol. 2008, Article ID 361705, 10 pages, 2008.

[14] Z. Ghahramani and M. Beal, "Propagation algorithms for variational Bayesian learning," in *Advances in Neural Information Processing Systems*, vol. 13, pp. 507–513, MIT Press, Cambridge, Mass, USA, 2000.

[15] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*, CRC Press, London, UK, 1996.

[16] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, New York, NY, USA, 2004.

[17] S. Chib, "Marginal likelihood from the gibbs output," *Journal of the Acoustical Society of America*, vol. 90, no. 432, pp. 1313–1321, 1995.

[18] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[19] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[20] M. V. Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete latent variable decomposition of counts data," in *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS '07)*, Vancouver, Canada, December 2007.

[21] C. S. Davis, "The computer generation of multinomial random variates," *Computational Statistics and Data Analysis*, vol. 16, no. 2, pp. 205–217, 1993.

[22] M. A. Tanner, *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, Springer, New York, NY, USA, 3rd edition, 1996.

[23] F. A. Quintana, J. S. Liu, and G. E. del Pino, "Monte Carlo EM with importance reweighting and its applications in random effects models," *Computational Statistics and Data Analysis*, vol. 29, no. 4, pp. 429–444, 1999.

[24] K. Butler and M. Stephens, "The distribution of a sum of binomial random variables," Tech. Rep. 467, Stanford University, Stanford, Calif, USA, April 1993, prepared for the Office of Naval Research.

[25] T. Minka, *Expectation propagation for approximate Bayesian inference*, Ph.D. thesis, MIT Media Lab, Cambridge, Mass, USA, 2001.

[26] O. Winther and K. B. Petersen, "Bayesian independent component analysis: variational methods and non-negative decompositions," *Digital Signal Processing*, vol. 17, no. 5, pp. 858–872, 2007.

[27] W. Buntine, "Variational extensions to EM and multinomial PCA," in *Proceedings of the 13th European Conference on Machine Learning (ECML '02)*, vol. 2430 of *Lecture Notes In Computer Science*, pp. 23–34, Springer, Helsinki, Finland, August 2002.

[28] A. T. Cemgil, P. Peeling, O. Dikmen, and S. Godsill, "Prior structures for time-frequency energy distributions," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '07)*, pp. 151–154, New Paltz, NY, USA, October 2007.

[29] P. Flajolet, "Singularity analysis and asymptotics of Bernoulli sums," *Theoretical Computer Science*, vol. 215, no. 1-2, pp. 371–381, 1999.

[30] P. Jacquet and W. Szpankowski, "Entropy computations via analytic depoissonization," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1072–1081, 1999.