Infinite Independent Components Analysis

by

David Knowles (JN)

Fourth-year undergraduate project in
Group F, 2006/2007

I hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signed: _____ Date: _____

Technical Abstract

An extension of Independent Components Analysis (ICA) is proposed where observed data $\mathbf{y}_t$ is modelled as a mixture, $\mathbf{G}$, of a potentially infinite number of hidden sources, $\mathbf{x}_t$. Whether a given source is active for a specific data point is specified by an infinite binary matrix, $\mathbf{Z}$, so

$$\mathbf{Y} = \mathbf{G}(\mathbf{Z} \odot \mathbf{X}) + \mathbf{E} \tag{1}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{X}$ and $\mathbf{Y}$ are concatenated matrices of $\mathbf{x}_t$ and $\mathbf{y}_t$ respectively, and $\mathbf{E}$ is Gaussian noise. We define four model variants which share the following priors:

$$\boldsymbol{\epsilon}_t \sim \mathcal{N}\left(0, \sigma_\epsilon^2 \mathbf{I}\right) \qquad \sigma_\epsilon^2 \sim \mathcal{IG}\left(a, b\right) \tag{2}$$

$$\mathbf{g}_k \sim \mathcal{N}\left(0, \sigma_G^2\right) \qquad \sigma_G^2 \sim \mathcal{IG}\left(c, d\right) \tag{3}$$

$$\mathbf{Z} \sim \mathcal{IBP}(\alpha, \beta) \qquad \alpha \sim \mathcal{G}\left(e, f\right) \tag{4}$$

where $\mathcal{G}\left(\cdot\right)$ denotes the Gamma distribution, $\mathcal{IG}\left(\cdot\right)$ the inverse Gamma distribution, and $\mathcal{IBP}(\cdot)$ the Indian Buffet Process (IBP). The variants differ in what source distribution and which version of the IBP is used:

	$x_{kt} \sim \mathcal{N}\left(0, 1\right)$	$x_{kt} \sim \mathcal{L}(1)$
$\beta = 1$	$isFA_1$	$iICA_1$
$\beta \sim \mathcal{G}\left(1, 2\right)$	$isFA_2$	$iICA_2$

where $\mathcal{L}(\cdot)$ is the Laplacian (bi-exponential) distribution. The IBP is a distribution over an infinite binary matrix $\mathbf{Z}$ found by taking the limit as $K \to \infty$ of the following generative model:

$$\pi_k | \alpha, \beta \sim \text{Beta}\left(\frac{\alpha\beta}{K}, \beta\right) \tag{5}$$

$$z_{kt} | \pi_k \sim \text{Bernoulli}(\pi_k) \tag{6}$$

In the one parameter IBP we fix $\beta = 1$. Through a stochastic process representation of this model we find the conditional probability of an element being 1 given all other elements is

$$P(z_{kt} = 1 | \mathbf{z}_{-kt}, \beta) = \frac{m_{k,-t}}{\beta + N - 1} \tag{7}$$

where $m_{k,-t} = \sum_{s \neq t} z_{ks}$. This facilities sampling of the elements of $\mathbf{Z}$.

We demonstrate Bayesian inference under each model variant using Markov Chain Monte Carlo (MCMC) methods: Gibbs sampling where possible and Metropolis-Hastings otherwise. We wish to infer the model parameters and hidden variables $\boldsymbol{\theta} = \{\mathbf{G}, \mathbf{X}, \mathbf{Z}, \sigma_e^2, \sigma_g^2, \alpha, \beta\}$ given observed data $\mathbf{Y}$. Gibbs sampling proceeds by sampling successively from the conditional distribution of one parameter given all others, i.e. $P(\theta_i|\mathbf{Y}, \boldsymbol{\theta}_{-i}) \propto P(\mathbf{Y}|\boldsymbol{\theta})P(\theta_i)$, by Baye's rule. Asymptotically this generates samples from the posterior $P(\boldsymbol{\theta}|\mathbf{Y})$.

The proposed algorithms are tested on 30 datasets generated randomly from our model, with $K = 6$ sources, $D = 200$ samples and $D = 7$ observed variables. Figure 1 shows boxplots of the Amari error for each algorithm alongside corresponding results for FastICA for comparison. The performance is superior to the standard FastICA algorithm.

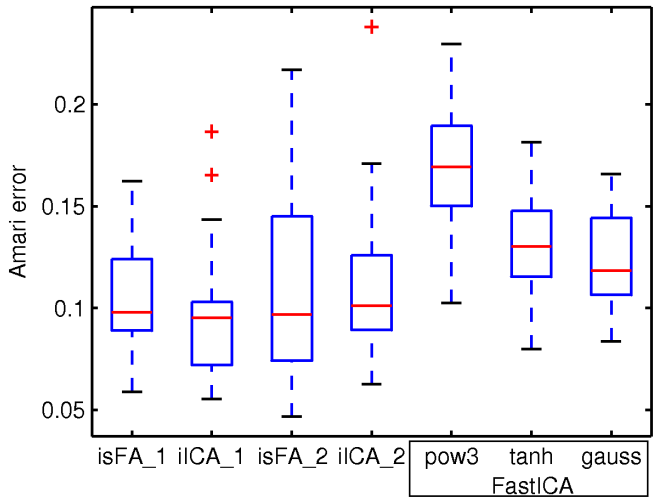


Figure 1: Boxplots of Amari errors for 30 synthetic data sets with $D = 7, N = 6, N = 100$ for each algorithm.

The algorithms successfully unmix artifically mixed audio sources, with one variant performing significantly better than FastICA and also being able to show when the sources are active. When applied to gene expression data from an ovarian cancer study the results are consistent with those in the literature. Finally the algorithms are applied to the growth rates of eleven FTSE100 companies. Hidden sources are inferred which account for the correlation across companies in the same industry.

Contents

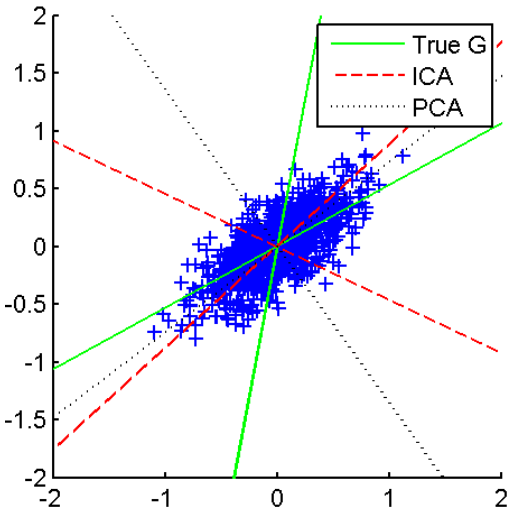
1 Introduction

Independent Components Analysis (ICA) is a model which explains observed data, $\mathbf{y}_t$ (dimension D) in terms of a linear superposition of independent hidden sources, $\mathbf{x}_t$ (dimension K), so

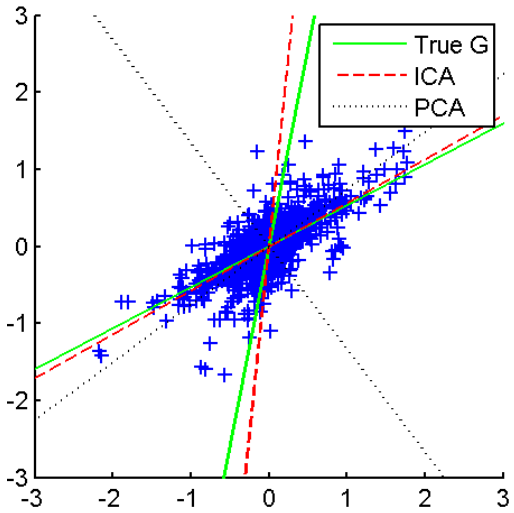
$$\mathbf{y}_t = \mathbf{G}\mathbf{x}_t + \boldsymbol{\epsilon}_t \tag{8}$$

where $\mathbf{G}$ is the mixing matrix and $\boldsymbol{\epsilon}_t$ is Gaussian noise. In the standard ICA model we assume $K = D$ and that there exists $\mathbf{W} = \mathbf{G}^{-1}$. Various algorithms for inferring $\mathbf{W}$ and $\mathbf{X}$ have been proposed, including steepest descent [4], covariant maximum likelihood [14], mixture of Gaussians based Independent Factor Analysis [2], constrast function based FastICA [12], and the cumultant based JADE [5]. It can be shown that most of these algorithms are equivalent to directly or indirectly maximising some measure of the non-gaussianity of the outputs. As a result, these algorithms will not work well with true Gaussian sources because the contrast functions become rotationally invariant. In fact most algorithms are designed to work best with heavy-tailed *super-Gaussian* source distributions which are common in nature, but some will also work well with light-tailed *sub-Gaussian* distributions. Figure 2 illustrates this dependence on the source distribution for a 2x2 mixture. FastICA is able to correctly calculate the mixing matrix in the case of heavy-tailed (Cauchy or Laplace) and light-tailed (uniform) source distributions, but not for Gaussian source distributions.

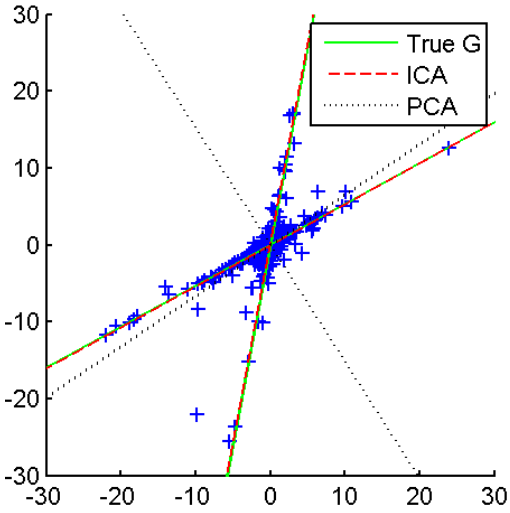
The assumption $K = D$ may be invalid, so Reversible Jump MCMC [20] could be used to infer K. Alternatively, Automatic Relevance Determination [15] suppresses irrelevant components. In this paper we propose a sparse implementation which allows a potentially infinite number of components and the choice of whether a hidden source is active for a data point. Although ICA is not a true time-series model it has been used successfully in analysing time-series data such as mixed audio (blind source separation) [2], stock returns [3], and electroencephalograms [16]. Note that attempts have been made to incorporate temporal learning into ICA, for example in [19]. It has also been applied to gene expression data [17], and it is this application that we choose for a demonstration.



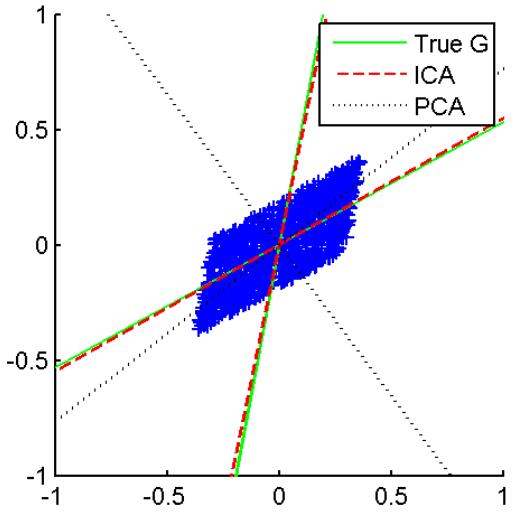
(a) Gaussian sources. The mixture is rotationally invariant so inference is unsuccessful.



(b) Laplacian sources. Slightly heavy tailed so inference is possible but inaccurate.



(c) Heavy tailed (Cauchy) sources. Accurate determination of directions: note PCA still fails because of orthogonal eigenvector constraint.



(d) Uniform sources. Light tailed so inference is possible.

Figure 2: ICA for different source distributions, showing the true mixing matrix directions and those inferred by ICA (using the FastICA algorithm) and PCA.

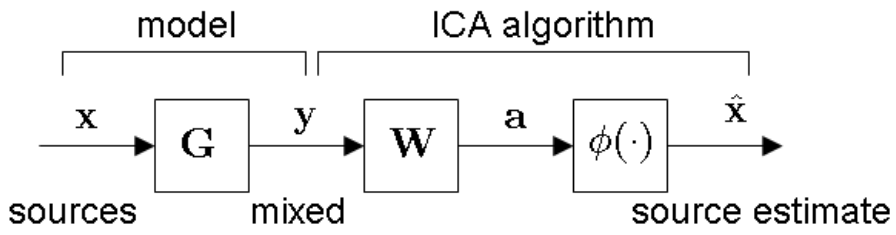


Figure 3: Neural network view of ICA. The sources $\mathbf{x}$ are estimated by a demixing matrix $\mathbf{W}$ and a non-linearity $\phi(\cdot)$.

1.1 *Infomax* Origins

ICA can be viewed as a blind source separation algorithm from an information theoretic perspective. The formulation of [4] attempts to maximise the mutual information $I(\hat{\mathbf{X}}, \mathbf{Y})$ of a single layer non-linear neural network, as shown in Figure 3. The mutual information is given by:

$$I(\hat{\mathbf{X}}, \mathbf{Y}) = H(\hat{\mathbf{X}}) - H(\hat{\mathbf{X}}|\mathbf{Y}) \tag{9}$$

where H is the differential entropy:

$$H(X) = -\int p(x) \log p(x) dx \tag{10}$$

For a system $\hat{\mathbf{x}} = \psi(\mathbf{y}) + \mathbf{e}$ where ψ is an invertible (i.e. monotonically increasing) transformation and $\mathbf{e}$ is additive noise, $H(\hat{\mathbf{X}}|\mathbf{Y}) = H(\mathbf{e})$. Thus the only component of $I(\hat{\mathbf{X}}, \mathbf{Y})$ that depends on the weights $\mathbf{W}$ is $H(\hat{\mathbf{X}})$: maximising the mutual information is equivalent to maximising the entropy of the output. The multivariate probability density function of $\hat{\mathbf{x}}$ can be written

$$f_{\hat{\mathbf{x}}}(\hat{\mathbf{x}}) = \frac{f_{\mathbf{y}}(\mathbf{y})}{|\mathbf{J}|} \tag{11}$$

where $|\mathbf{J}|$ is the absolute value of the Jacobian of the transformation. Thus the entropy of the output

$$H(\hat{\mathbf{X}}) = -\int f_{\hat{\mathbf{x}}}(\hat{\mathbf{x}}) \ln f_{\hat{\mathbf{x}}}(\hat{\mathbf{x}}) d\hat{\mathbf{x}} \tag{12}$$

$$= -\int f_{\mathbf{y}}(\mathbf{y}) \ln \frac{f_{\mathbf{y}}(\mathbf{y})}{|\mathbf{J}|} d\mathbf{y} \tag{13}$$

$$= E[\ln |\mathbf{J}|] + H(\mathbf{Y}) \tag{14}$$

Since $H(\mathbf{Y})$ is fixed our aim becomes to maximise $E[\ln |\mathbf{J}|]$, which is the volume in $\hat{\mathbf{x}}$ that points in $\mathbf{y}$ are mapped to. Thus we attempt to spread the output as much as possible. The Jacobian for a single data point is

$$\mathbf{J} = |\mathbf{W}| \prod_i \phi'(a_i) \tag{15}$$

$$\Rightarrow \ln |\mathbf{J}| = \ln |\mathbf{W}| + \sum_i \ln \phi'(a_i) \tag{16}$$

Maximising this expression with respect to the mixture weights with sigmoidal $\phi(\cdot)$ using steepest descent gives Bell and Sejowski's algorithm [4]:

$$\Delta \mathbf{W} \propto \mathbf{W}^{-T} + (\mathbf{1} - 2\hat{\mathbf{x}})\mathbf{y}^T \tag{17}$$

FastICA [12] is a more efficient algorithm derived within the mutual information framework but which optimizes a constrast function which is an appromixation to the negentropy, a measure of non-gaussianity, of the outputs.

1.2 From *infomax* to Maximum Likelihood

We will now show that this information maximisation approach is equivalent to maximum likelihood estimation in the no noise limit, following Mackay [14]. The

likelihood function for a single data point is then

$$P(\mathbf{y}^{(n)}|\mathbf{G}) = \int P(\mathbf{y}^{(n)}|\mathbf{G}, \mathbf{x}^{(n)}) P(\mathbf{x}^{(n)}) d\mathbf{x}^{(n)} \tag{18}$$

$$= \int \prod_j \delta(y_j^{(n)} - (\mathbf{G}\mathbf{x}^{(n)})_j) \prod_i p_i(x_i^{(n)}) d\mathbf{x}^{(n)} \tag{19}$$

$$= \frac{1}{|\mathbf{G}|} \prod_i p_i(a_i) \tag{20}$$

where $\mathbf{a} = \mathbf{G}^{-1}\mathbf{y}$ and we have used the matrix analogy to the scalar identity $\int \delta(y - gx) f(x) dx = \frac{1}{g} f\left(\frac{y}{g}\right)$. Now let $\hat{\mathbf{W}} = \mathbf{G}^{-1}$ we have

$$\ln P(\mathbf{y}^{(n)}|\mathbf{G}) = \ln |\hat{\mathbf{W}}| + \sum_i \ln p_i(a_i) \tag{21}$$

We see this is equivalent to Equation (16) if we identify $\hat{\mathbf{W}} = \mathbf{W}$ and use the interpretation that the non-linearity $\phi_i(\cdot)$ should approximate the cdf of p_i.

$$\phi_i(a_i) = \int_{-\infty}^{a_i} p_i(x) dx \tag{22}$$

$$\Rightarrow \phi_i'(a_i) = p_i(a_i) \tag{23}$$

This equivalence was shown independently in [6] by showing both methods minimise the KL divergence between the true and estimated densities of $\mathbf{x}$. In [14] Equation (21) is maximised using an approximation of Newton's algorithm ($\Delta w = -H^{-1} \nabla \ln P(\mathbf{y}^{(n)}|\mathbf{G})$, where H is the Hessian matrix) to give a faster, simpler, covariant (i.e. scale invariant) algorithm which does not require inversion of the mixing matrix at each step:

$$\Delta \mathbf{W} \propto (\mathbf{I} - \hat{\mathbf{x}}\mathbf{a}^T)\mathbf{W} \tag{24}$$

1.3 Bayesian inference

Maximum Likelihood estimation is known to be associated with various problems, primarily overfitting and local maxima. A full Bayesian treatment would attempt to calculate the posterior over the model parameters and hidden variables, $\mathcal{H}$, given

the observed data, $\mathcal{V}$, i.e.

$$P(\mathcal{H}|\mathcal{V}) = \frac{P(\mathcal{V}|\mathcal{H})\mathcal{H}}{P(\mathcal{V})} \tag{25}$$

This calculation is in general not analytically tractable, so we need to use an approximation. Two main approaches exist: variational methods and Monte Carlo methods.

1.3.1 Variational methods

Variational methods try to find an approximation $q(\mathcal{H})$ to the true distribution $P(\mathcal{H}|\mathcal{V})$. From Jensen's inequality, a rigorous lower bound on the marginalised log likelihood can be obtained:

$$\ln \int P(\mathcal{H}, \mathcal{V}) d\mathcal{H} \geq \int q(\mathcal{H}) \ln \frac{P(\mathcal{H}, \mathcal{V})}{q(\mathcal{H})} d\mathcal{H} \tag{26}$$

The difference between the lower bound and the true marginalised log likelihood is given by the Kullback-Lieber divergence between the approximation $q(\mathcal{H})$ and the true distribution $P(\mathcal{H}|\mathcal{V})$:

$$KL(P \parallel q) = \int q(\mathcal{H}) \ln \frac{P(\mathcal{H}|\mathcal{V})}{q(\mathcal{H})} d\mathcal{H} \tag{27}$$

The approximation q must be choosen to be sufficiently simple to allow analytic and computational tractability but flexible enough to allow the bound in Equation (26) to be tight. Often a factored distribution is used which assumes independence between the various model parameters. Both [13] and [7] develop Bayesian ICA models using this approach with mixture of Gaussian source distributions (as in IFA [2]), but the nature of the IBP prior used in this paper makes Monte Carlo methods the natural choice for this project.

1.3.2 Monte Carlo methods

For all but the simplest models finding the posterior over the parameters is not analytically possible so we attempt to approximate it by drawing samples using MCMC, methods which take a random walk whose stationary distribution is the

target distribution. We can then find expectations using the unbiased estimator:

$$\int g(\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{V})d\boldsymbol{\theta} \approx \frac{1}{S}\sum_{s=1}^{S} g(\boldsymbol{\theta}^{(s)}), \qquad \boldsymbol{\theta}^{(s)} \sim P(\boldsymbol{\theta}|\mathcal{V}) \tag{28}$$

The simplest MCMC method is the Metropolis-Hastings algorithm, which requires a proposal distribution $Q(\theta';\theta)$. We draw a new sample θ' from this distribution, and accept it with probability $\min(1, r_{\theta\to\theta'})$ where

$$r_{\theta\to\theta'} = \frac{P(\theta'|\mathcal{V})Q(\theta;\theta')}{P(\theta|\mathcal{V})Q(\theta';\theta)} \tag{29}$$

It can easily be shown that this proposal ensures detailed balance:

$$Q(\theta';\theta)P(\theta|\mathcal{V}) = Q(\theta;\theta')P(\theta'|\mathcal{V}) \tag{30}$$

Detailed balance in turn ensures that $P(\theta|\mathcal{V})$ is a stationary distribution of the Markov chain, since by integrating with respect to θ' we find

$$P(\theta|\mathcal{V}) = \int P(\theta'|\mathcal{V})Q(\theta;\theta')d\theta' \tag{31}$$

The problem with Metropolis-Hastings is how to choose the proposal distribution Q: too broad and most of the proposas will be rejected, too narrow and it will take too long for the Markov chain to explore the distribution. There is no such arbitrary choice to make using Gibbs sampling, an MCMC algorithm where we successively sample each parameter, θ_k from its conditional distribution given all the other parameters, $\boldsymbol{\theta}_{-k}$, i.e. $P(\theta_k|\boldsymbol{\theta}_{-k},\mathcal{V})$. It can be shown [8] that this procedure generates samples from the parameter posterior, $P(\boldsymbol{\theta}|\mathcal{V})$. Where possible we will use Gibbs sampling rather than Metropolis-Hastings, but for some steps this is not possible.

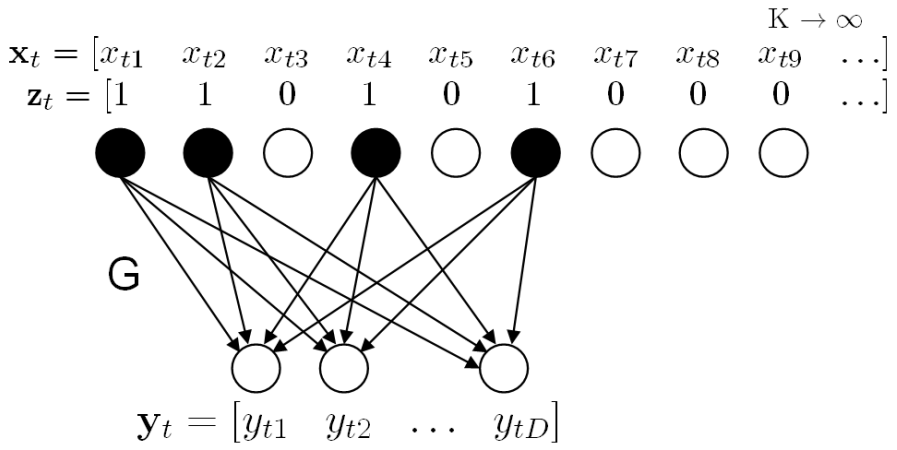


Figure 4: Diagrammatic representation of Infinite ICA model

2 The Model

We define a binary vector $\mathbf{z}_t$ which acts as a mask on $\mathbf{x}_t$. Element z_{kt} specifies whether hidden source k is active for data point t. Thus

$$\mathbf{Y} = \mathbf{G}(\mathbf{Z} \odot \mathbf{X}) + \mathbf{E} \tag{32}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{E}$ are concatenated matrices of $\mathbf{x}_t$, $\mathbf{y}_t$, $\mathbf{z}_t$ and $\boldsymbol{\epsilon}_t$ respectively. We allow a potentially infinite number of hidden sources, so that $\mathbf{Z}$ has infinitely many rows, although only a finite number will have non-zero entries. We assume Gaussian noise with variance σ_ϵ^2, which is given an inverse Gamma prior, i.e.

$$P(\sigma_\epsilon^2|a,b) = \mathcal{IG}\left(\sigma_\epsilon^2; a, b\right) = \frac{(\sigma_\epsilon^2)^{-(a+1)}}{b^a \Gamma(a)} \exp\left(-\frac{1}{b\sigma_\epsilon^2}\right) \tag{33}$$

We define two variants based on the prior for x_{kt}: *infinite sparse Factor Analysis* (isFA) has a unit Gaussian prior; *infinite Independent Components Analysis* (iICA) has a Laplacian(1) prior. Varying the variance is redundant because we infer the variance of the mixture weights. The prior on the elements of $\mathbf{G}$ is Gaussian with variance σ_G^2, which is given an inverse Gamma prior. We define the prior on $\mathbf{Z}$ using the Indian Buffet Process with parameter α (and later β) as described in Section 2.1 and in more detail in [11]. We place Gamma priors on α and β.

All four variants share

$$\boldsymbol{\epsilon}_t \sim \mathcal{N}\left(0, \sigma_\epsilon^2 \mathbf{I}\right) \qquad \sigma_\epsilon^2 \sim \mathcal{IG}\left(a, b\right) \tag{34}$$

$$\mathbf{g}_k \sim \mathcal{N}\left(0, \sigma_G^2\right) \qquad \sigma_G^2 \sim \mathcal{IG}\left(c, d\right) \tag{35}$$

$$\mathbf{Z} \sim \mathcal{IBP}(\alpha, \beta) \qquad \alpha \sim \mathcal{G}\left(e, f\right) \tag{36}$$

The differences between the variants are summarised here.

	$x_{kt} \sim \mathcal{N}(0,1)$	$x_{kt} \sim \mathcal{L}(1)$
$\beta = 1$	$isFA_1$	$iICA_1$
$\beta \sim \mathcal{G}(1,2)$	$isFA_2$	$iICA_2$

2.1 Defining a distribution on an infinite binary matrix

2.1.1 Start with a finite model.

We derive our distribution on $\mathbf{Z}$ by defining a finite K model and taking the limit as $K \to \infty$. We then show how the infinite case corresponds to a simple stochastic process.

We have N data points and K hidden sources. Recall that z_{kt} of matrix $\mathbf{Z}$ tells us whether hidden source k is active for time t. We assume that the probability of a source k being active is π_k, and that the sources are generated independently. We find

$$P(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{k=1}^{K} \prod_{t=1}^{N} P(z_{kt}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1-\pi_k)^{N-m_k} \tag{37}$$

where $m_k = \sum_{t=1}^{N} z_{kt}$ is the number of data points for which source k is active. The inner term of the product is a binomial distribution, so we choose the conjugate Beta(r,s) distribution for π_k. For now we take $r = \frac{\alpha}{K}$ and $s = 1$, where α is the strength parameter of the IBP. The model is defined by

$$\pi_k|\alpha \sim \text{Beta}\left(\frac{\alpha}{K}\right) \tag{38}$$

$$z_{kt}|\pi_k \sim \text{Bernoulli}(\pi_k) \tag{39}$$

Due to the conjugacy between the binomial and beta distributions we are able to

integrate out π to find

$$P(\mathbf{Z}) = \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})} \tag{40}$$

where $\Gamma(.)$ is the Gamma function.

2.1.2 Take the infinite limit.

By defining a scheme to order the non-zero rows of $\mathbf{Z}$ (see [11]) we can take $K \to \infty$ and find

$$P(\mathbf{Z}) = \frac{\alpha^{K_+}}{\prod_{h>0} K_h!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!} \tag{41}$$

where K_+ is the number of active features, $H_N = \sum_{j=1}^{N} \frac{1}{j}$ is the N-th harmonic number, and K_h is the number of rows whose entries correspond to the binary number h.

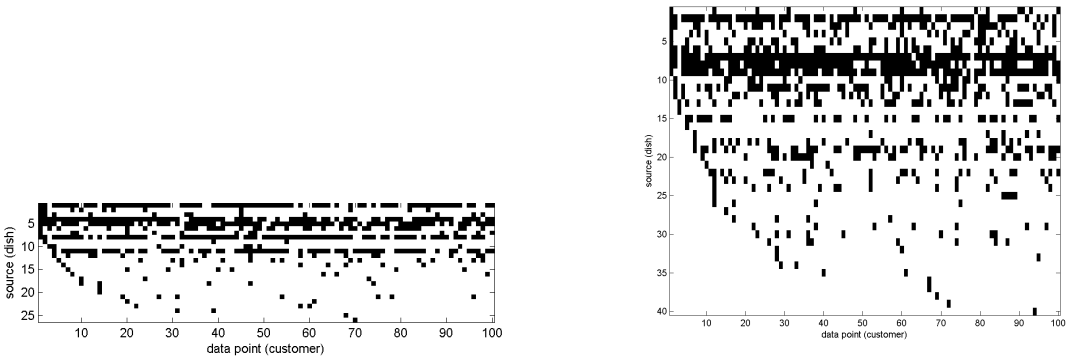
2.1.3 Go to an Indian Buffet.

This distribution corresponds to a simple stochastic process, the Indian Buffet Process. Consider a buffet with a seemingly infinite number of dishes (hidden sources) arranged in a line. The first customer (data point) starts at the left and samples Poisson(α) dishes. The ith customer moves from left to right sampling dishes with probability $\frac{m_k}{i}$ where m_k is the number of customers to have previously sampled that dish. Having reached the end of the previously sampled dishes, he tries Poisson($\frac{\alpha}{i}$) new dishes. Figure 5 shows two draws from the IBP for two different values of α.

If we apply the same ordering scheme to the matrix generated by this process as for the finite model, we recover the correct exchangeable distribution. Since the distribution is exchangeable with respect to the customers we find by considering the last customer that

$$P(z_{kt} = 1 | \mathbf{z}_{-kt}) = \frac{m_{k,-t}}{N} \tag{42}$$

where $m_{k,-t} = \sum_{s \neq t} z_{ks}$, which is used in sampling $\mathbf{Z}$. By exchangeability and



(a) $\alpha = 4$ (b) $\alpha = 8$

Figure 5: Draws from the one parameter IBP for two different values of α.

considering the first customer, the number of active sources for a data point follows a Poisson(α) distribution, and the expected number of entries in $\mathbf{Z}$ is $N\alpha$. We also see that the number of active features, $K_+ = \sum_{t=1}^{N} \text{Poisson}(\frac{\alpha}{t}) = \text{Poisson}(\alpha H_N)$.

2.1.4 Two parameter generalisation.

A problem with the one parameter IBP is that the number of features per object, α, and the total number of features, $N\alpha$, are both controlled by α and cannot vary independently. Under this model, we cannot tune how likely it is for features to be shared across objects. To overcome this restriction we follow [9], introducing β, a measure of the feature *repulsion*. The ith customer now samples dish k with probability $\frac{m_k}{\beta+i-1}$ and samples Poisson($\frac{\alpha\beta}{\beta+i-1}$) new dishes.

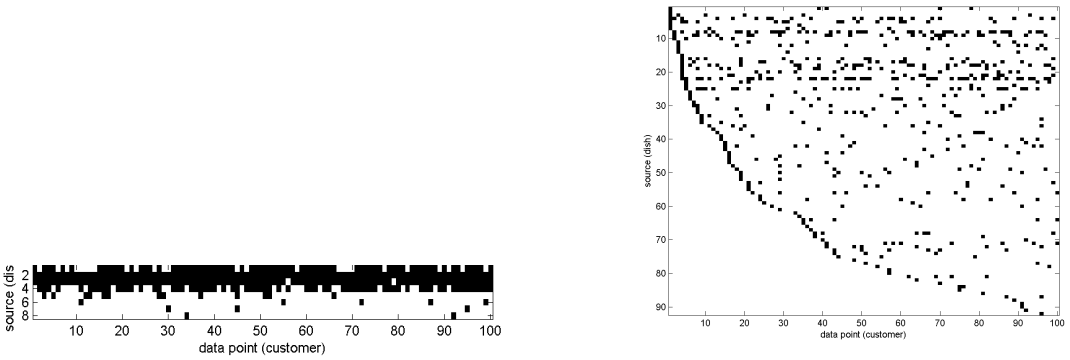
Figure 6 shows draws from the two parameter IBP for two different values of β. For $\beta < 1$ we get increased sharing of sources amongst data points, as in Figure 6(a), and for $\beta > 1$ we get reduced sharing, as in Figure 6(b).

Following the same thread as for the one parameter IBP, we find

$$P(z_{kt} = 1 | \mathbf{z}_{-kt}, \beta) = \frac{m_{k,-t}}{\beta + N - 1} \tag{43}$$

The marginal probability of $\mathbf{Z}$ becomes

$$P(\mathbf{Z}|\alpha, \beta) = \frac{(\alpha\beta)^{K_+}}{\prod_{h>0} K_h!} \exp\{-\alpha H_N(\beta)\} \prod_{k=1}^{K_+} B(m_k, N - m_k + \beta) \tag{44}$$



(a) $\alpha = 8, \beta = 0.1$ (b) $\alpha = 8$

Figure 6: Draws from the one parameter IBP for two different values of α.

where C is a constant with respect to α and β, and $H_N(\beta) = \sum_{j=1}^{N} \frac{\beta}{\beta+j-1}$. The expected overall number of active features is now $\bar{K}_+ = \alpha H_N(\beta)$. We will derive all our results for the two parameter case because it is straightforward to recover the one parameter case by setting $\beta = 1$.

2.2 Stick Breaking Construction

An alternative representation of the IBP has recently been proposed for the one-parameter IBP in [21], which allows a slice sampling method to be derived allowing potentially faster mixing in the non-conjugate source distribution case. Again we start with the finite case, but now construct a decreasing ordering of the π_k of Equation (38): $\pi_{(1)} > \pi_{(2)} > \cdots > \pi_{(K)}$. In [21] it is shown that $\mu_{(k)}$ obey the following equation:

$$\nu_{(k)} \sim \text{Beta}\,(\alpha, 1)\pi_{(k)} = \nu_{(k)}\mu_{(k-1)} = \prod_{l=1}^{k} \nu_{(l)} \tag{45}$$

The analogy we use is as follows. We start with a stick of length one, and break off a length $\nu_{(1)}$, and record its length as $\pi_{(1)}$. At iteration k, we break off a length $\nu_{(k)}$ relative to the remaining length, and record its length as $\pi_{(k)}$.

3 Inference

Given the observed data $\mathbf{Y}$, we wish to infer the hidden sources $\mathbf{X}$, which sources are active $\mathbf{Z}$, the mixing matrix $\mathbf{G}$, and all hyperparameters. We use Gibbs sampling, but with Metropolis-Hastings (MH) steps for β and sampling new features. We draw samples from the marginal distribution of the model parameters given the data by successively sampling the conditional distributions of each parameter in turn, given all other parameters. Pseudocode for the overall algorithm can be found in Appendix C.

3.1 Likelihood function

The likelihood function for a specific data-point, t, is

$$P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_t, \mathbf{z}_t, \sigma_\epsilon^2) = \mathcal{N}\left(\mathbf{y}_t; \mathbf{G}(\mathbf{z}_t \circ \mathbf{x}_t), \sigma_\epsilon^2 \mathbf{I}\right) \tag{46}$$

$$= \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \exp\left\{-\frac{1}{2\sigma_\epsilon^2}(\mathbf{y}_t - \mathbf{G}(\mathbf{z}_t \circ \mathbf{x}_t))^T(\mathbf{y}_t - \mathbf{G}(\mathbf{z}_t \circ \mathbf{x}_t))\right\} \tag{47}$$

since $\mathbf{y}_t = \mathbf{G}(\mathbf{z}_t \circ \mathbf{x}_t) + \boldsymbol{\epsilon}_t$.

Since the data-points are assumed i.i.d. the likelihood function for the whole dataset is just

$$P(\mathbf{Y}|\mathbf{G}, \mathbf{X}, \mathbf{Z}) = \prod_{t=1}^{N} P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_t, \mathbf{z}_t) \tag{48}$$

$$= \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{ND}{2}}} \exp\left\{-\frac{1}{2\sigma_\epsilon^2} \operatorname{tr}(\mathbf{Y} - \mathbf{G}(\mathbf{Z} \circ \mathbf{X}))^T(\mathbf{Y} - \mathbf{G}(\mathbf{Z} \circ \mathbf{X}))\right\} \tag{49}$$

3.2 Hidden sources.

We sample each element of $\mathbf{X}$ for which $z_{kt} = 1$. We denote the k-th column of $\mathbf{G}$ by $\mathbf{g}_k$ and $\boldsymbol{\epsilon}_t|_{z_{kt}=0}$ by $\boldsymbol{\epsilon}_{-kt}$. For isFA we find (see Appendix A.1.1) this is a

Gaussian:

$$P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) \propto P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_t, \mathbf{z}_t, \sigma_\epsilon^2)P(x_{kt}) \tag{50}$$

$$= \mathcal{N}\left(x_{kt}; \frac{\mathbf{g}_k^T \boldsymbol{\epsilon}_{-kt}}{\sigma_\epsilon^2 + \mathbf{g}_k^T \mathbf{g}_k}, \frac{\sigma_\epsilon^2}{\sigma_\epsilon^2 + \mathbf{g}_k^T \mathbf{g}_k}\right) \tag{51}$$

For iICA we find (see Appendix A.1.2) a piecewise Gaussian distribution, which it is possible to sample from analytically given the inverse of the Gaussian cdf.

$$P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) = \begin{cases} \frac{B_+}{A}\mathcal{N}\left(x_{kt}; \mu_+, \sigma^2\right) & x_{kt} > 0 \\ \frac{B_-}{A}\mathcal{N}\left(x_{kt}; \mu_-, \sigma^2\right) & x_{kt} < 0 \end{cases} \tag{52}$$

where $\mu_\pm, \sigma, B_\pm$ and A are defined in Equation (96).

3.3 Active sources.

To sample $\mathbf{Z}$ we first define the ratio of conditionals, r

$$r = \frac{P(z_{kt} = 1|\mathbf{G}, \mathbf{X}_{-kt}, \mathbf{Y}, \mathbf{Z}_{-kt})}{P(z_{kt} = 0|\mathbf{G}, \mathbf{X}_{-kt}, \mathbf{Y}, \mathbf{Z}_{-kt})} \tag{53}$$

$$= \underbrace{\frac{P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1, \sigma_\epsilon^2)}{P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 0, \sigma_\epsilon^2)}}_{r_l} \underbrace{\frac{P(z_{kt} = 1|\mathbf{z}_{-kt})}{P(z_{kt} = 0|\mathbf{z}_{-kt})}}_{r_p} \tag{54}$$

so that $P(z_{kt} = 1|\mathbf{G}, \mathbf{X}_{-kt}, \mathbf{Y}, \mathbf{Z}_{-kt}) = \frac{r}{r+1}$. From Equation (43) we find the ratio of priors is $r_p = \frac{m_{k,-t}}{\beta+N-1-m_{k,-t}}$. The likelihood evaluated with $z_{kt} = 0$ is

$$P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 0) = \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \exp\left\{-\frac{\boldsymbol{\epsilon}_{-kt}^T \boldsymbol{\epsilon}_{-kt}}{2\sigma_\epsilon^2}\right\} \tag{55}$$

where $\boldsymbol{\epsilon}_{-kt} = \boldsymbol{\epsilon}_t|_{z_{kt}=0}$ is the error vector $\boldsymbol{\epsilon}_t$ evaluated with $z_{kt} = 0$. To find $P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1)$ we must marginalise over all possible values of x_{kt}.

$$P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1, \sigma_\epsilon^2) = \int P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_t, \mathbf{z}_{-kt}, z_{kt} = 1, \sigma_\epsilon^2)P(x_{kt})\mathsf{d}x_{kt} \tag{56}$$

This result clearly depends on the form of the prior on x_{kt}.

For isFA the integrand is that of Equation (50) so we are able to use the same

results.

$$P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1, \sigma_\epsilon^2) = \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \frac{1}{\sqrt{2\pi}} \int \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{\epsilon}_{-kt} - \mathbf{g}_k x_{kt}\|^2 - \frac{x_{kt}^2}{2}\right\} \mathsf{d}x_{kt} \tag{57}$$

Using Equation (50) and integrating we find the ratio of likelihoods $r_l = \sigma \exp\left\{\frac{\mu^2}{2\sigma^2}\right\}$ where $\sigma^2 = \frac{\sigma_\epsilon^2}{\sigma_\epsilon^2 + \mathbf{g}_k^T \mathbf{g}_k}$ and $\mu = \frac{\mathbf{g}_k^T \boldsymbol{\epsilon}_{-kt}}{\sigma_\epsilon^2 + \mathbf{g}_k^T \mathbf{g}_k}$

For iICA we have

$$P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1, \sigma_\epsilon^2) = \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \frac{1}{2} \int \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{\epsilon}_{-kt} - \mathbf{g}_k x_{kt}\|^2 - |x_{kt}|\right\} \mathsf{d}x_{kt} \tag{58}$$

Completing the square and integrating above and below zero we find the ratio of likelihoods is

$$r_l = \sigma\sqrt{\frac{\pi}{2}} \left[F(0; \mu_+, \sigma) \exp\left\{\frac{\mu_+^2}{2\sigma^2}\right\} + (1 - F(0; \mu_-, \sigma)) \exp\left\{\frac{\mu_-^2}{2\sigma^2}\right\}\right] \tag{59}$$

where μ_-, μ_+, σ are as defined in Equation (96).

If z_{kt} is changed from 0 to 1 we interleave a sampling of x_{kt}. If it is changed from 1 to 0 we set $x_{kt} = 0$.

3.4 Creating new features.

$\mathbf{Z}$ is a matrix with infinitely many rows, but only the non-zero rows can be held in memory. However, the zero rows still need to be taken into account. Let κ_t be the number of rows of $\mathbf{Z}$ which contain 1 only in column t, i.e. the number of features which are active only at time t. Figure 7 illustrates κ_t for a sample $\mathbf{Z}$ matrix.

New features are proposed by sampling κ_t with a MH step. We propose a move $\xi \to \xi^*$ with probability $J(\xi^*|\xi)$, following [18], we set to be equal to the prior on ξ^*. This move is accepted with probability $\min(1, r_{\xi \to \xi^*})$ where

$$r_{\xi \to \xi^*} = \frac{P(\xi^*|\text{rest})J(\xi|\xi^*)}{P(\xi^*|\text{rest})J(\xi^*|\xi)} = \frac{P(\text{rest}|\xi^*)P(\xi^*)P(\xi)}{P(\text{rest}|\xi)P(\xi)P(\xi^*)} = \frac{P(\text{rest}|\xi^*)}{P(\text{rest}|\xi)} \tag{60}$$

By this choice $r_{\xi \to \xi^*}$ becomes the ratio of likelihoods. From the IBP the prior for

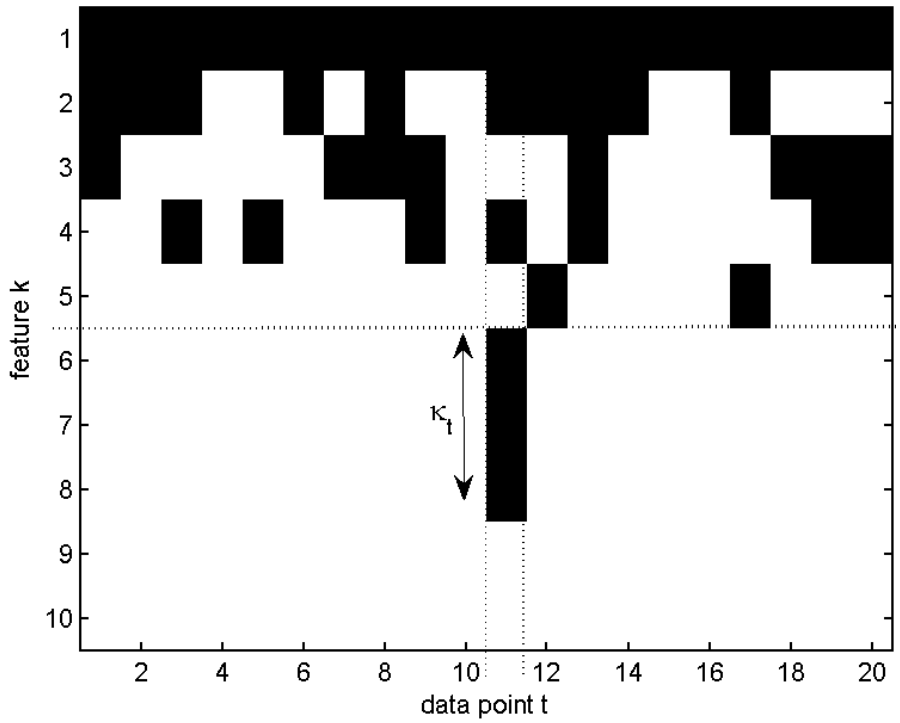


Figure 7: A diagram to illustrate the definition of κ_t.

κ_t is

$$P(\kappa_t|\alpha) = \text{Poisson}(\frac{\alpha\beta}{\beta + N - 1}) \tag{61}$$

For isFA we find it is possible to integrate out $\mathbf{x}'_t$, the new elements of $\mathbf{x}_t$, but not $\mathbf{G}'$, the new columns of $\mathbf{G}$, so our proposal ξ^* includes not only κ_t^* but also the new columns $\mathbf{G}^*$. We now marginalise over $\mathbf{x}'_t$.

$$P(\mathbf{y}_t|\mathbf{G}, \mathbf{G}', \mathbf{x}_t, \mathbf{z}_t, \kappa_t, \sigma_\epsilon^2) = \int P(\mathbf{y}_t|\mathbf{G}, \mathbf{G}', \mathbf{x}_t, \mathbf{x}'_t, \mathbf{z}_t) P(\mathbf{x}'_t) \mathsf{d}\mathbf{x}'_t \tag{62}$$

$$= (2\pi\sigma_\epsilon^2)^{-\frac{D}{2}} (2\pi)^{-\frac{\kappa_t}{2}} \int \exp\left\{ -\frac{1}{2\sigma_\epsilon^2} \|\boldsymbol{\epsilon}_t - \mathbf{G}'\mathbf{x}'_t\|^2 - \frac{\mathbf{x}_t'^T \mathbf{x}'_t}{2} \right\} \mathsf{d}\mathbf{x}'_t \tag{63}$$

Integrating and using Equation (60) we have

$$r_{\xi \to \xi^*} = |\boldsymbol{\Lambda}|^{-\frac{1}{2}} \exp\left(\frac{1}{2} \boldsymbol{\mu}^T \boldsymbol{\Lambda} \boldsymbol{\mu} \right) \tag{64}$$

where $\mathbf{\Lambda} = \mathbf{I} + \frac{\mathbf{G}^{*T}\mathbf{G}^*}{\sigma_\epsilon^2}$ and $\mathbf{\Lambda}\boldsymbol{\mu} = \frac{1}{\sigma_\epsilon^2}\mathbf{G}^{*T}\boldsymbol{\epsilon}_t$.

For iICA it is not possible to integrate out $\mathbf{x}'_t$ or $\mathbf{G}'$ so these are included in the proposal, $\xi = \{\mathbf{G}', \mathbf{x}'_t, \kappa_t\}$. From Equation (60) we find

$$r_{\xi \to \xi^*} = \exp\left\{-\frac{1}{2\sigma_\epsilon^2}\mathbf{x}_t'^T\mathbf{G}^{*T}(\mathbf{G}^*\mathbf{x}'_t - 2\boldsymbol{\epsilon}_t)\right\} \tag{65}$$

3.5 Mixture weights.

We sample the columns $\mathbf{g}_k$ of $\mathbf{G}$. We denote the kth row of $\mathbf{Z} \odot \mathbf{X}$ by ${\mathbf{s}_k}^T$. We have $P(\mathbf{g}_k|\mathbf{G}_{-k}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \sigma_\epsilon^2, \sigma_G^2) \propto P(\mathbf{Y}|\mathbf{G}, \mathbf{X}, \mathbf{Z}, \sigma_\epsilon^2)P(\mathbf{g}_k|\sigma_G^2)$. The total likelihood function has exponent

$$-\frac{1}{2\sigma_\epsilon^2}\operatorname{tr}(\mathbf{E}^T\mathbf{E}) = -\frac{1}{2\sigma_\epsilon^2}(({\mathbf{s}_k}^T\mathbf{s}_k)(\mathbf{g}_k^T\mathbf{g}_k) - 2\mathbf{g}_k^T\mathbf{E}|_{\mathbf{g}_k=0}) + \text{const} \tag{66}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{G}(\mathbf{X} \odot \mathbf{Z})$. This is shown in Appendix A.2. We thus find the conditional of $\mathbf{g}_k$, $P(\mathbf{g}_k|\mathbf{G}_{-k}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \sigma_\epsilon^2, \sigma_G^2) \propto P(\mathbf{Y}|\mathbf{G}, \mathbf{X}, \mathbf{Z}, \sigma_\epsilon^2)P(\mathbf{g}_k|\sigma_G^2) = \mathcal{N}(\mathbf{g}_k; \boldsymbol{\mu}, \mathbf{\Lambda})$ by comparing coefficients in $\mathbf{g}_k$, where

$$\mathbf{\Lambda} = \left(\frac{{\mathbf{s}_k}^T\mathbf{s}_k}{\sigma_\epsilon^2} + \frac{1}{\sigma_G^2}\right)\mathbf{I}_{D\times D} \tag{67}$$

$$\boldsymbol{\mu} = \frac{\sigma_G^2}{{\mathbf{s}_k}^T\mathbf{s}_k\sigma_G^2 + \sigma_\epsilon^2}\mathbf{E}|_{\mathbf{g}_k=0}\mathbf{s}_k \tag{68}$$

where $\mathbf{E}|_{\mathbf{g}_k=0}$ is $\mathbf{Y} - \mathbf{G}(\mathbf{X} \odot \mathbf{Z})$ evaluated with $\mathbf{g}_k = 0$.

3.6 Learning the noise level.

We allow the model to learn the noise level σ_ϵ^2. Applying Bayes' rule we find (see Appendix A.3)

$$P(\sigma_\epsilon^2|\mathbf{E}, a, b) \propto P(\mathbf{E}|\sigma_\epsilon^2)P(\sigma_\epsilon^2|a, b) = \mathcal{IG}\left(\sigma_\epsilon^2; a + \frac{ND}{2}, \frac{b}{1 + \frac{b}{2}\operatorname{tr}(\mathbf{E}^T\mathbf{E})}\right) \tag{69}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{G}(\mathbf{X} \odot \mathbf{Z})$. We draw samples from the inverse Gamma distribution by taking the reciprocal of samples drawn from a Gamma distribution with the same parameters.

3.7 Inferring the scale of the data.

For sampling σ_G^2 the conditional prior on $\mathbf{G}$ acts as the likelihood term since the likelihood itself is independent of σ_G^2 given $\mathbf{G}$ so we find (see Appendix A.4)

$$P(\sigma_G^2|\mathbf{G}, c, d) \propto P(\mathbf{G}|\sigma_G^2)P(\sigma_G^2|c, d) = \mathcal{IG}\left(\sigma_G^2; c + \frac{DK}{2}, \frac{d}{1 + \frac{d}{2}\operatorname{tr}(\mathbf{G}^T\mathbf{G})}\right) \quad (70)$$

3.8 IBP parameters.

We infer the IBP strength parameter α. The conditional prior on $\mathbf{Z}$, given by Equation (44), acts as the likelihood term so we find (see Appendix A.5)

$$P(\alpha|\mathbf{Z}, \beta) \propto P(\mathbf{Z}|\alpha, \beta)P(\alpha) = \mathcal{G}\left(\alpha; K_+ + e, \frac{f}{1 + fH_N(\beta)}\right) \quad (71)$$

We sample β by a Metropolis-Hasting's step with acceptance probability $\min(1, r_{\beta\to\beta^*})$. By Equation (60) we know that setting the proposal distribution equal to the prior, i.e. $J(\beta^*|\beta) = P(\beta^*) = \mathcal{G}(1, 1)$, results in $r_{\beta\to\beta^*}$ being equal to the ratio of likelihoods, in this case $\frac{P(Z|\alpha,\beta^*)}{P(Z|\alpha,\beta)}$ as given in Equation (44).

3.9 Slice Sampler

As a result of the stick-breaking construction of the IBP, an alternative inference scheme can be used which where the π_k are not integrated out. We define an auxiliary variable s which is effectively an adaptive truncution level for π. We sample

$$s|\mathbf{Z}, \pi_{(1:\infty)} \sim \text{Uniform}[0, \pi^*] \quad (72)$$

where π^* is the smallest value of π for all the active features (i.e. those which are active for at least one data point). Given s, the posterior distribution of $\mathbf{Z}$ is

$$P(\mathbf{Z}|\text{rest}, s, \pi_{(1:\infty)}) \propto P(\mathbf{Z}|\text{rest}, \pi_{(1:\infty)})P(s|\mathbf{Z}, \pi_{(1:\infty)}) \quad (73)$$

$$\propto \begin{cases} P(\mathbf{Z}|\text{rest}, \pi_{(1:\infty)}) & 0 \le s \le \pi^* \\ 0 & \text{otherwise} \end{cases} \quad (74)$$

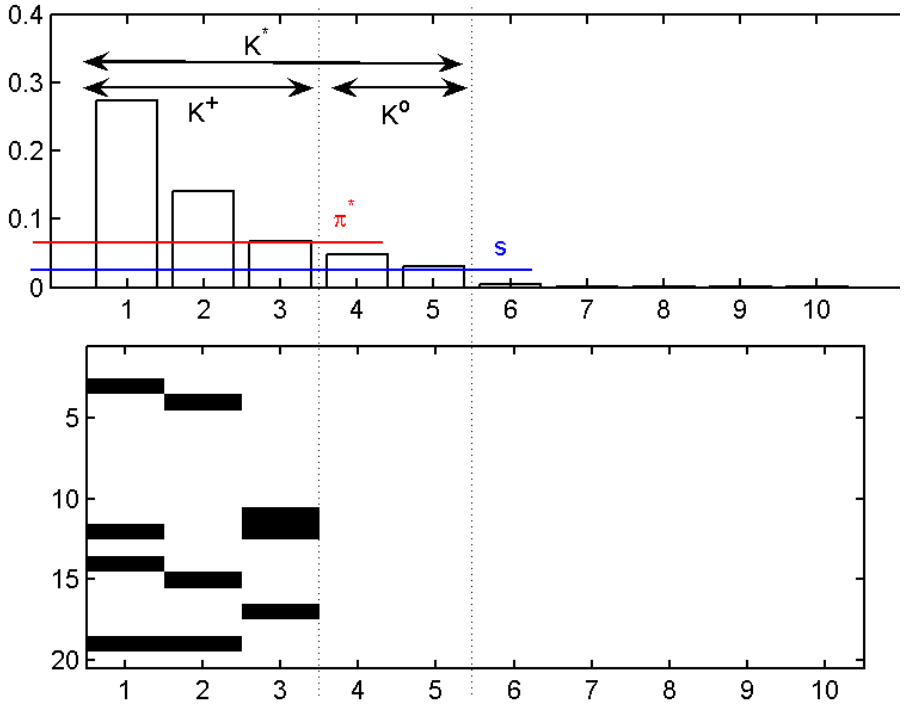


Figure 8: *Top*: A draw of $\pi_{(k)}$ using the stick breaking construction with $\alpha = 1.5$. *Bottom*: A corresponding draw of $\mathbf{Z}$ with $N = 20$, shown transposed.

Thus all rows of $\mathbf{Z}$ for which $\pi_k < s$ must be zero. We denote the maximal feature index with $\pi_{(k)} > s$ by K^*, and note that we need only update features for $k < K^*$. Our computational representation for the slice sampler will include up to feature $K^\dagger$. We may have to pad our representation with inactive features to ensure $K^* < K^\dagger$. Figure 8 shows a draw of $\pi_{(k)}$ for $\alpha = 1.5, N = 20$, with the corresponding $\mathbf{Z}$ matrix. The significance of the parameters π^*, s, K^* is also illustrated.

In [21] it is shown that the new stick lengths can be drawn iteratively from

$$p(\pi_{(k)}|\pi_{(k-1)}, z_{>k,:} = 0) \tag{75}$$

$$\propto \begin{cases} \exp\left\{\alpha \sum_N^{i=1} \frac{1}{i}(1-\pi_{(k)})^i\right\} \pi_{(k)}^{\alpha-1}(1-\pi_{(k)})^N, & 0 \le \pi_{(k)} \le \pi_{(k-1)} \\ 0, & \text{otherwise} \end{cases} \tag{76}$$

We sample from this distribution using Adaptive Rejection Sampling [10] because it is log-concave in $\log \pi_{(k)}$.

To sample z_{kt} we now use the formula:

$$P(z_{kt} = 1|\text{rest}) \propto P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1)P(s|\mathbf{Z}, \pi_{(1:\infty)})P(z_{kt} = 1|\pi_{(k)}) \quad (77)$$

$$\propto P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{z}_{-kt}, z_{kt} = 1)\frac{\pi_{(k)}}{\pi^*} \quad (78)$$

Note that π^* is a function of $\mathbf{Z}$ and so may change if z_{kt} changes.

We sample $\pi_{(k)} \forall k = 1, \cdots, K^\dagger - 1$ from

$$p(\pi_{(k)}|\text{rest}) \propto \begin{cases} \pi_{(k)}^{m_k - 1}(1 - \pi_{(k)})^{N - m_k} & \pi_{(k+1)} \leq \pi_{(k)} \leq \pi_{(k-1)} \\ 0, & \text{otherwise} \end{cases} \quad (79)$$

This distribution can also be sampled using ARS. We sample $\pi_{(K^\dagger)}$ using Equation (75) with $k = K^\dagger$.

3.10 Semi-ordered Slice Sampling

It turns out that we only really need to enforce the ordering of $\pi_{(k)}$ on the inactive features. We will store K^+ active, unordered features, and K° inactive, ordered ones. For the active features, π_k^+

$$\pi_k^+|z_{k,:} \sim \text{Beta}(m_k, N - m_k + 1) \quad (80)$$

For the inactive features, $\pi_{(k)}^\circ$ are sampled from Equation (75). Our sampler will now take the following steps:

1. Sample s
2. Generate new features until $\pi_{(K^\circ+1)}^\circ < s$
3. Sample the active and inactive features as before
4. Remove inactive features
5. Sampling π_k^+ from their conditional Equation (80)

4 Results

4.1 Evaluation

How best to evaluate the algorithm depends on whether the data under consideration is synthetic or real world. In the synthetic case we can compare inferred $\mathbf{G}$, $\mathbf{X}$ and $\mathbf{Z}$ to known ground truth data, using the standard *Amari error* which is both scale and permutation invariant in $\mathbf{X}$. In the case of real world data we do not ground truth values to compare to, so we instead attempt evaluate predictive performance on test data.

4.1.1 Amari error

Let $\mathbf{S} = \mathbf{X} \odot \mathbf{Z}$, then in the no noise limit we have

$$\mathbf{Y} = \mathbf{GS} \tag{81}$$

However, there are two indetermancies: scaling by a diagonal matrix $\mathbf{\Sigma}$ and permutation by a matrix $\mathbf{\Pi}$ of the sources:

$$\mathbf{Y} = \mathbf{G\Sigma}^{-1}\mathbf{\Pi}^{-1}\mathbf{\Sigma\Pi S} \tag{82}$$

If we now write the inferred sources $\hat{\mathbf{S}} = \mathbf{MS}$ and mixing matrix $\hat{\mathbf{G}} = \mathbf{GM}^{-1}$ then we see that we have recovered the sources optimally if $\mathbf{M} = \mathbf{\Sigma\Pi}$. Solving for $\mathbf{M}$ we find

$$\mathbf{M} = (\hat{\mathbf{S}}\mathbf{S}^T)(\mathbf{S}\mathbf{S}^T)^{-1} \tag{83}$$

We can now define the Amari error in terms of $\mathbf{M}$.

$$E = \frac{1}{2KK' - K' - K}\left(\sum_{i=1}^{K'}\left(\frac{\sum_{j=1}^{K}|M_{ij}|}{\max_k |M_{ik}|} - 1\right) + \sum_{j=1}^{K}\left(\frac{\sum_{i=1}^{K}|M_{ij}|}{\max_k |M_{kj}|} - 1\right)\right) \tag{84}$$

where K is the true number of sources and K' is the inferred number. The Amari error is normally defined for $K = K'$ but since we infer the number of sources we may have $K \neq K'$. The Amari error is the sum over rows and columns of the deviation from there only being one main entry per column, normalised so that the

maximum error is 1. For perfect recovery the Amari error will be zero.

4.1.2 Cross validation

To evaluate predictive performance, we can separate the data into training data, $\mathcal{D}_1$, and test data, $\mathcal{D}_2$. As part of the algorithm we evaluate $L_{2|1} = \log P(\mathcal{D}_2|\mathcal{D}_1) \approx \int \log P(\mathcal{D}_2|\theta) P(\theta|\mathcal{D}_1) d\theta$, where θ is the set of all parameters of the model ($\mathbf{X}_2$ and $\mathbf{Z}_2$ for the test data must be also be inferred).

4.1.3 Predictive performance

A more "honest" way to assess the algorithm's performance is to look at how well it predicts a future value of the data by evaluating $p(\mathbf{y}|\mathbf{Y})$ where $\mathbf{y}$ is the new data point. This is similar to Leave Out One Cross Validation, and is a computationally intensive but rigorous performance index. We take $S = 20$ independent samples of $\theta = \{\mathbf{G}, \mathbf{X}, \mathbf{Z}, \sigma_\epsilon\}$ from the algorithm and use the approximation

$$\begin{aligned} p(\mathbf{y}|\mathbf{Y}) &= \int p(\mathbf{y}, \theta|\mathbf{Y}) d\theta \\ &= \int p(\mathbf{y}|\theta) p(\theta|\mathbf{Y}) d\theta \\ &\approx \frac{1}{S} \sum_{s=1}^{S} p(\mathbf{y}|\theta^{(s)}), \qquad\qquad \theta^{(s)} \sim p(\theta|\mathbf{Y}) \end{aligned} \tag{85}$$

We now need to evaluate $p(\mathbf{y}|\theta)$ which we do by integrating out the hidden sources for this data point, $\mathbf{x}$ and sampling over which of these sources are active, $\mathbf{z}$. Firstly integrating over $\mathbf{x}$:

$$p(\mathbf{y}|\mathbf{z}, \theta) = \int p(\mathbf{y}|\mathbf{z}, \theta) p(\mathbf{x}) d\mathbf{x} \tag{86}$$

We let $\mathbf{A}_{ij} = \mathbf{G}_{ij} z_j$ (no sum over j) so that $\mathbf{G}(\mathbf{x} \odot \mathbf{z}) = \mathbf{A}\mathbf{x}$. For isFA we have

$$\begin{aligned} p(\mathbf{y}|\mathbf{z}, \theta) &= \int \frac{1}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_\epsilon^2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2\right\} \frac{1}{(2\pi)^{\frac{K}{2}}} \exp\left\{-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right\} d\mathbf{x} \\ &= \frac{|\mathbf{\Sigma}|}{(2\pi\sigma_\epsilon^2)^{\frac{D}{2}}} \exp\left\{\frac{1}{2}\boldsymbol{\mu}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu} - \frac{1}{2\sigma_\epsilon^2}\mathbf{y}^T\mathbf{y}\right\} \end{aligned} \tag{87}$$

where $\mathbf{\Sigma}^{-1} = \frac{1}{\sigma_\epsilon^2}\mathbf{A}^T\mathbf{A} + \mathbf{I}$ and $\mathbf{\Sigma}^{-1}\boldsymbol{\mu} = \frac{1}{\sigma_\epsilon^2}\mathbf{A}^T\mathbf{y}$. We can then marginalise over $\mathbf{z}$

$$
\begin{aligned}
p(\mathbf{y}|\theta) &= \sum_{\mathbf{z}} p(\mathbf{y}|\mathbf{z},\theta)P(\mathbf{z}|\mathbf{Z},\alpha,\beta) \\
&\approx \frac{1}{U}\sum_{u=1}^{U} p(\mathbf{y}|\mathbf{z}^{(u)},\theta), \qquad\qquad \mathbf{z}^{(u)} \sim P(\mathbf{z}|\mathbf{Z},\alpha,\beta) \qquad (88)
\end{aligned}
$$

where $P(\mathbf{z}|\mathbf{Z},\alpha,\beta)$ is given by the IBP. The pseudocode for this evaluation is shown in Algorithm 1.

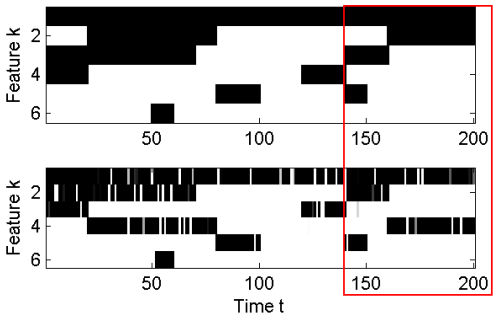
Algorithm 1 Calculate the predictive performance $p(\mathbf{y}|\mathbf{Y})$

```
for s = 1, ··· , S do
    for u = 1, ··· , U do
        Draw z ~ P(z^(u)|Z, α, β) using Equation (43)
        Calculate p(y|z^(u), θ^(s)) from Equation (87)
    end for
    Calculate p(y|θ^(s)) from Equation (88)
end for
Calculate p(y|Y) from Equation (85)
```

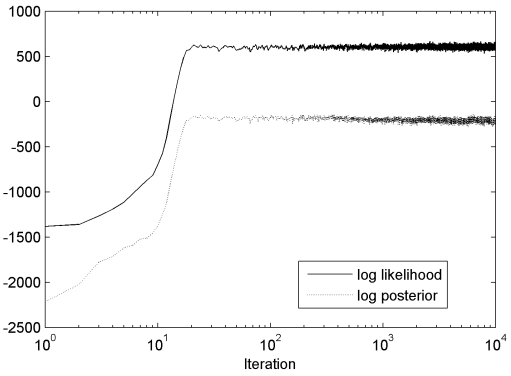
4.2 Synthetic data

The algorithms were tested on synthetic data with $N = 200$, $D = 8$, $\mathbf{X}$ and $\mathbf{G}$ drawn from their priors, and the $\mathbf{Z}$ shown in Figure 9(a), with $K = 6$ hidden sources. Although $\mathbf{G}$ was drawn from its prior we ensured it was not too close to being singular by restricting its condition number to be less than 5. This ensures the problem is well conditioned and it is at least theoretically possible to recover the soures. The average $\mathbf{Z}$ inferred by iICA$_2$ is shown in Figure 9(a). We find the sources within an arbitrary ordering. The gaps in the inferred $\mathbf{Z}$ are a result of inferring $z_{kt} = 0$ where $x_{kt} = 0$. Figure 9(b) shows the variation of the log likelihood and posterior over a long, 10^4 iteration run, and Figure 10 shows the main parameters' variation over a 1000 iteration run.

Figure 11(a) shows $L_{2|1}$ (see Section 4.1.2) for each variant: the predictive power of iICA appears to be greater than that of isFA, and the two-parameter IBP versions show worse generalization. Figure 11(b) shows the permutation-invariant normalised Amari error [1] for each variant on synthetic data with a



(a) *Top:* True $\mathbf{Z}$. *Bottom:* Inferred $\mathbf{Z}$. Red box denotes test data.



(b) Plot of the log likelihood and posterior for the duration of the iICA_2 run.

Figure 9: True and inferred $\mathbf{Z}$ and algorithm convergence.

Gaussian or Laplacian prior on $\mathbf{X}$. As expected, the more appropriate algorithm for the structure of the data performs best: isFA performs better on Gaussian than Laplacian data; and visa versa for iICA. Interestingly, iICA seems significantly worse at dealing with normally distributed data than isFA is at coping with heavy tailed data. This suggests that isFA is more robust to non-standard distributions than iICA. The two parameter IBP variants of both algorithms actually perform no better than the one parameter versions: $\beta = 1$ happens to be almost optimal for the synthetic $\mathbf{Z}$ used, so the one-parameter versions are not penalised by this restriction. Even the apparent improvement of iICA_2 over iICA_1 on Gaussian data is in fact because it allowed two inferred hidden sources to account for the greater variance of the true hidden source 1.

To compare the algorithm's performance to standard ICA we ran the one-parameter variants (isFA_1 and iICA_1) and three FastICA variants (using the *pow3, tanh* and *gauss* non-linearities) on 30 sets of randomly generated data. We did this for both Gaussian and Laplacian source distributions, the results of which are shown in Figure 12. Figure 12(a) shows the results when the synthetic data has Gaussian source distributions. Both isFA_1 and iICA_1 perform significantly better on the sparse synthetic data than any of the FastICA variants, which is unsurprisingly as we do not expect FastICA to recover Gaussian sources. The median average performance is very similar for both, although the iICA_1 performance is more variable with both more very low error results and more high error outliers. Figure 12(b) shows the results when the synthetic data has Laplacian source dis-

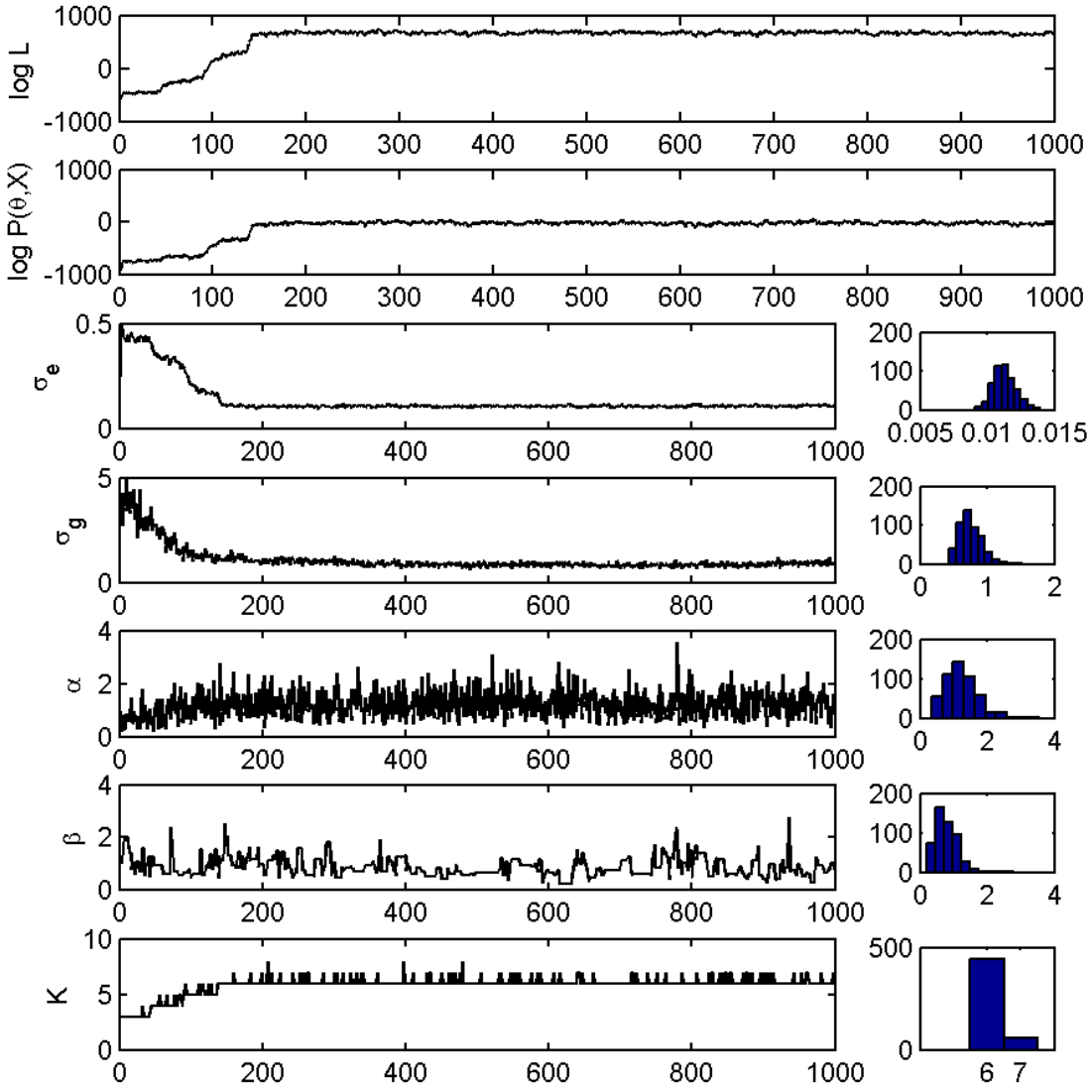
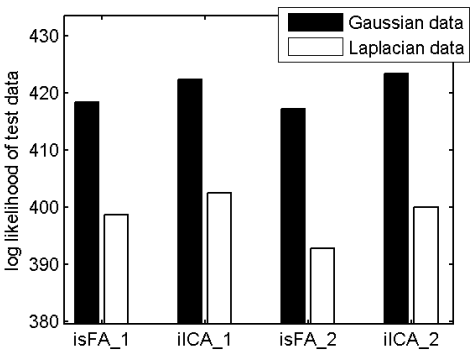
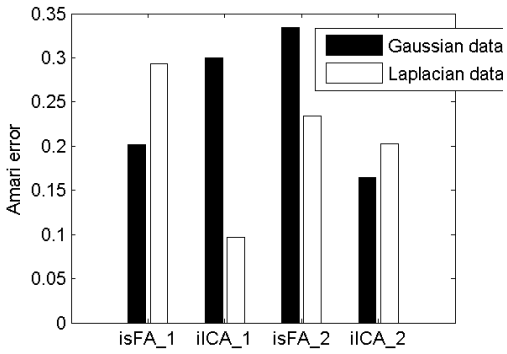


Figure 10: A typical isFA$_2$ run on synthetic data.



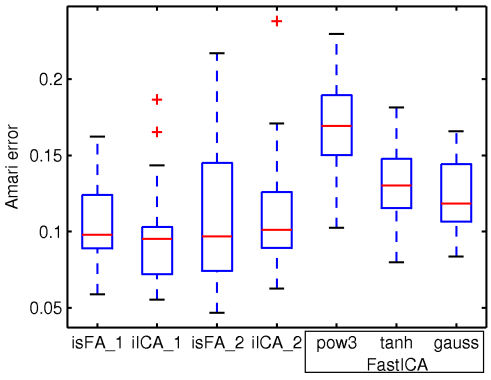
(a) Log likelihood of test data for each variant.



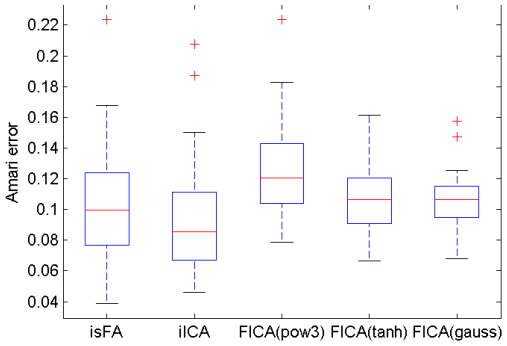
(b) Amari error for the four algorithm variants.

Figure 11: Results on Gaussian or Laplacian synthetic data.

tributions. As expected the FastICA performance is much improved because the sources are heavy tailed. However, isFA_1 and iICA_1 still perform better because they correctly model the sparse nature of the data. As expected iICA_1 outperforms isFA_1 in this case because it uses the correct source distribution.



(a) Gaussian sources.

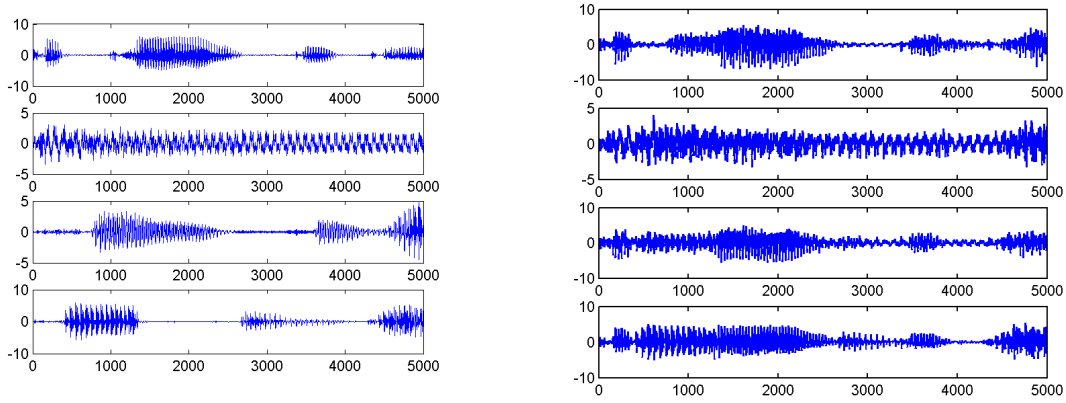


(b) Laplacian sources.

Figure 12: Boxplots of Amari errors for 30 synthetic data sets with $D = 7, N = 6, N = 100$ analysed using isFA_1, iICA_1 and FastICA algorithm variants. The red line shows the median, the box the interquartile range, the whiskers the extend of the remaining data, and the red crosses are outliers.

4.3 Audio data

We now apply the algorithms to a slightly more realistic dataset: an artificially mixed set of four audio sources, three speech and one music, with $N = 5000$ samples. Artificially mixing the sources prevents real world artifacts such as echoes which would have unpredictable effects on the algorithm performance. The original and mixed signals are shown in Figure 13.

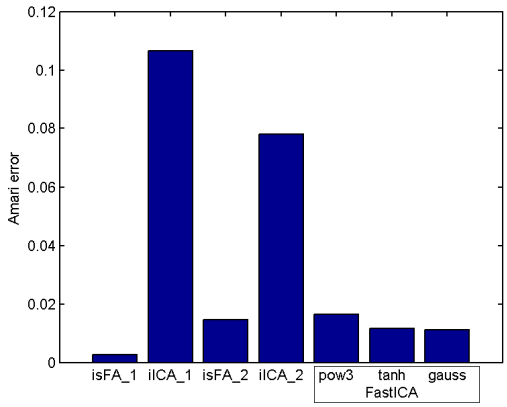


(a) Original sources.

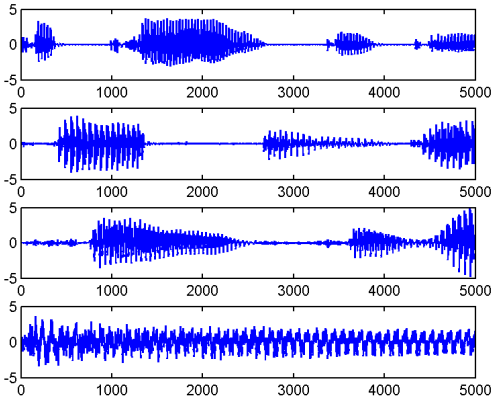
(b) Mixed signals.

Figure 13: Artificially mixed audio dataset.

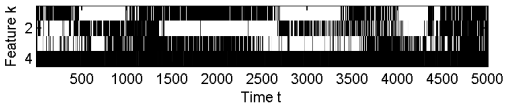
Each algorithm variant was run on this dataset for 1000 iterations, and the average source matrix $\mathbf{X}$ calculated for the last 500 iterations. The Amari error was then calculated for each, and is shown in Figure 14(a). The error is smallest for isFA$_1$, and slightly greater for isFA$_2$, roughly inline with the FastICA performance. The superior performance of isFA$_1$ over the FastICA algorithm is because our model naturally models the periods when the sources are inactive. The iICA variants perform much worse (although on an absolute scale an Amari error of 0.1 still implies the sources were recovered), implying that this data is not well modelled by the Laplacian source distribution assumption. Figure 14(b) and 14(c) and shows the sources $\mathbf{X}$ and average $\mathbf{Z}$ recovered respectively by isFA$_1$. The agreement (within permutation and scaling) of the inferred and true sources is excellent, and $\mathbf{Z}$ quite accurately represents when the sources were active.



(a) Amari error for each algorithm run on the audio dataset.



(b) Recovered sources.



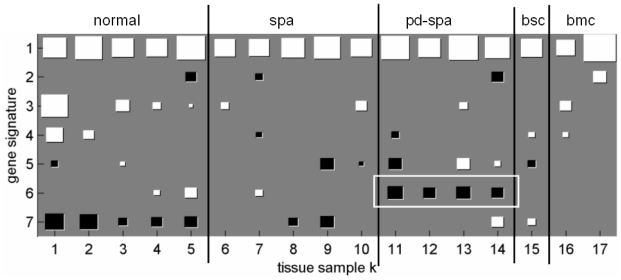
(c) Infered **Z** matrix.

Figure 14: Output of isFA_1 algorithm on audio dataset.

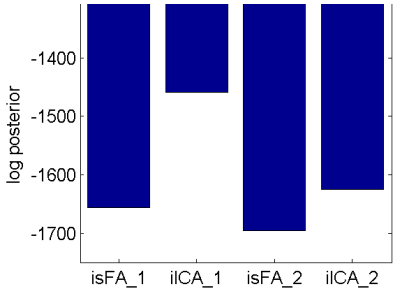
4.4 Gene expression data

We now apply our model to the microarray data from an ovarian cancer study [17], which represents the expression level of $D = 172$ genes across $N = 17$ tissue samples. These include 5 normal ovary, 5 serous papillary adenocarcinoma (spa), 4 poorly differentiated serous papillary adenocarcinoma (pd-spa), 1 benign serous cystadenoma (bsc), and 2 benign mucinous cystadenoma (bmc). ICA was applied to this dataset in [17]. The performance of our four variants on this data for 5000 iteration runs is compared in Figure 15(b). iICA_1 appears to perform best, producing the inferred **X** shown in Figure 15(a). Gene signature (hidden source) 1 is expressed across all the tissue samples, accounted for genes shared by all the samples. Signature 7 is specific to the pd-spa tisue type. This is consistent with that found in [17], with the same top 3 genes. Such tissue type dependent signatures could be used for observer independent classification. Signatures such as 5 which is differentially expressed across the pd-spa samples could help subclassify tissue types. Tissue sample 1 is pre-menopausal, which is detected by gene signature 3. Miskin notes the absence of any inferred gene signature to define the spa tissue type. This may be due to misclassification in the original data set, or due to

variation of genes and processes not included in the microarray.



(a) Hinton diagram of the gene signature expression level for each tissue sample.



(b) Mean log posterior for each algorithm variant.

Figure 15: Application to ovarian cancer data set.

For microarray studies a control measurement l_t is made for each array, t, to account for variation in the amount of solution available. Therefore we define our observed data $y_{dt} = \frac{m_{dt}}{l_t}$ where m_{dt} is the experimental measurement. We maintain our assumption of Gaussian noise in $\mathbf{Y}$, although this is potentially a poor assumption since $\mathbf{y}_t$ is formed as the ratio of two experimental measurements.

4.5 Financial data

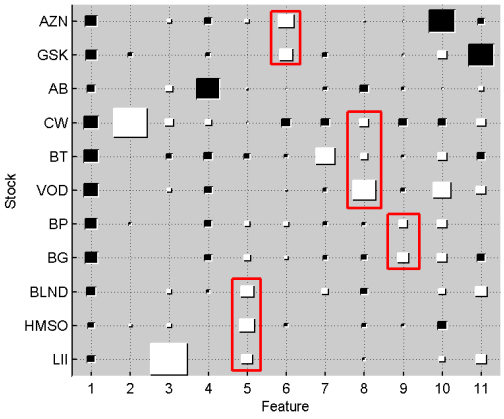
We now consider the application of these algorithms to stock market data. We used historical price data from Yahoo! Finance for ten FTSE100 companies considered to be potentially correlated:

Symbol	Stock	Industry
AZN	Astrazeneca	Pharmaceutical
GSK	Glaxo	Pharmaceutical
AB	Alliance Boots	Pharma/consumer
CW	Cable & Wireless	Telecoms
BT	BT Group	Telecoms
VOD	Vodafone	Telecoms
BP	BP Group	Energy
BG	BG Group	Energy
BLND	Brit Land Co Reit	Property
HMSO	Hammerson Reit	Property
LII	Liberty Int Reit	Property

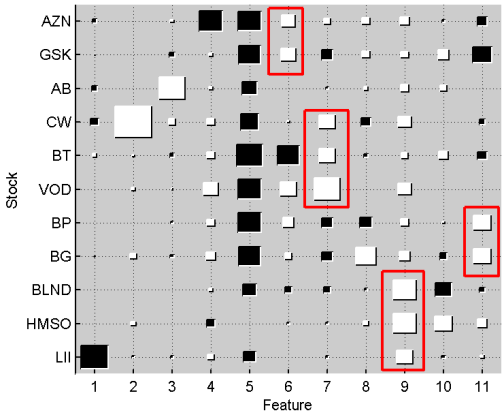
We collected daily closing prices, p_t, adjusted for dividends and splits, from 1st January 2002 to 29th December 2006, a total of $N = 1298$ samples. To make this data stationary we transform the data under the assumption of exponential growth:

$$y_t = \log \frac{p_t}{p_{t-1}} \tag{89}$$

The raw and transformed data is shown in Appendix B. We ran each algorithm variant for 2000 iterations and then calculated the predictive performance. This showed iICA$_2$ to best model the data, so we show the mean average $\mathbf{G}$ matrix for this and FastICA in Figure 16.



(a) Hinton diagram of the average mixing matrix, $\mathbf{G}$, for iICA$_2$ applied to the financial dataset.



(b) Hinton diagram of the mixing matrix for FastICA (pow3) applied to the financial dataset.

Figure 16: Application to financial data set.

Figure 16(a) shows the results for iICA$_2$. The red rectangles highlight the hidden features which account for the correlation between stocks in the same industry. Feature 1 is universally expressed across these stocks: this feature would probably be found across all FTSE100 stocks, and accounts for the shared effects. Feature 6 is expressed by the two pharmaceutical companies, feature 8 by the telecoms companies, feature 9 by the energy companies, and feature 5 by the property company. These results are very satisfying: the algorithm is able to find underlying driving forces of the various industries. An interesting further investigation would be to compare these inferred driving forces to factors that would be expected to control

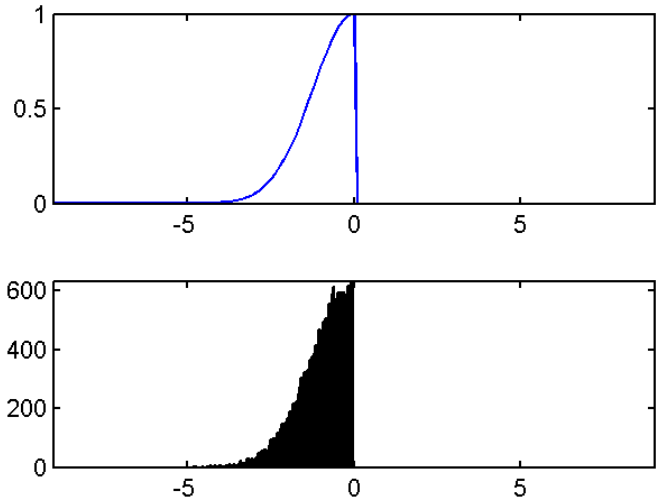


Figure 17: *Top*: Half Gaussian distribution. *Bottom*: 20000 samples drawn from this distribution using ARS.

a particular industry: for example the oil price for energy companies. Figure 16(b) shows the corresponding mixing matrix inferred by FastICA(pow3). We again find industry specific hidden features, but they are not as distinct in this case: for example feature 6 which is primarily expressed by the pharmaceutical companies also affects Vodafone! Therefore our sparse model may provide a more useful model for the data.

5 Slice sampling

To test the semi-ordered stick breaking construction it is necessary first to assess the Adaptive Rejection Samplng (ARS) algorithm I implemented following [10]. It was necessary to develop this algorithm as the only available Matlab implementation of ARS uses a less efficient variant which does not use the derivative of the log of the pdf being sampled. To test this algorithm it is used to draw sampling from a half-Gaussian, as shown in Figure 17. The agreement between the pdf and the samples is good: the algorithm operates correctly.

The semi-ordered stick breaking algorithm has convergence problems. The number of features K tends to grow unreasonably large for the data, and even when when the correct $\mathbf{Z}$ matrix is obtained it is easily disrupted by the addition of new features. When a new feature is created, the parameters associated with it (the

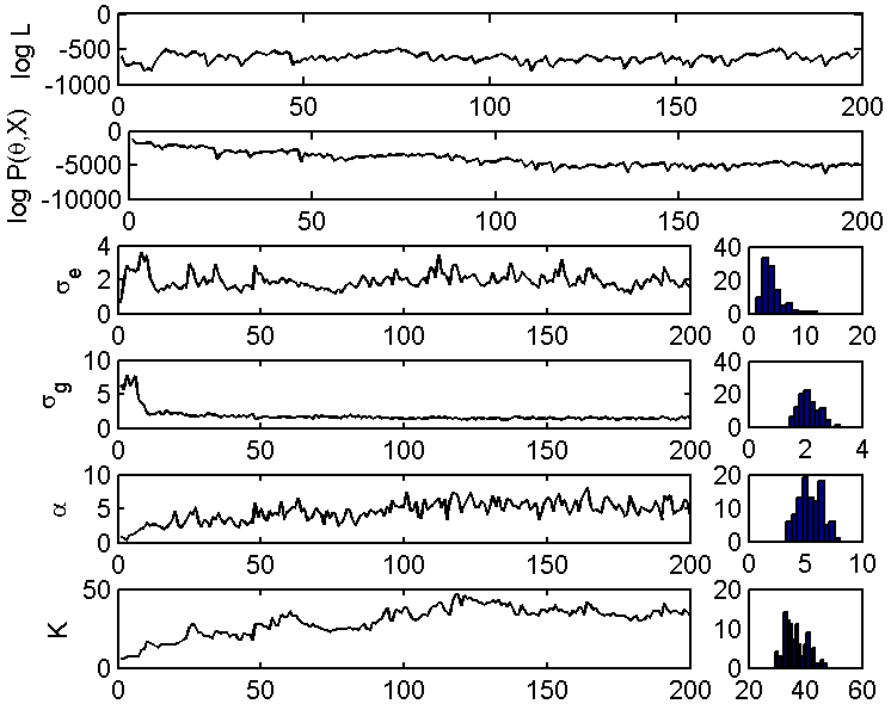


Figure 18: The semi-ordered slice sampler fails to converge on a toy data set with just two sources, infering a phantom 40 sources. The mixing of this MCMC method is unsuccessful.

new rows of $\mathbf{X}$ and columns of $\mathbf{G}$) are drawn from their priors. The $\mathbf{Z}$ matrix is then sampled with these random values of $\mathbf{X}$ and $\mathbf{G}$ held fixed, which greatly disrupts the distribution. The subsequent sampling of $\mathbf{G}$ and $\mathbf{X}$ is affected by the disruption to $\mathbf{Z}$, and the algorithm begins to diverge. In the worst case scenario a situation is reached where features are active simply to cancel each other out: indeed the fact that running this variant on synthetic data with just two sources resulted in an inferred K of around 40 (see Figure 18) shows that this algorithm does not perform correctly. To alleviate these problems I have tried changing the order of the sampling (for example, sampling the new features once before resampling old features) which has improved the performance but not entirely rectified the issue.

6 Conclusion

In this paper we have defined the Infinite Sparse FA and Infinite ICA models using a distribution over the infinite binary matrix $\mathbf{Z}$ corresponding to the one or two-parameter Indian Buffet Process. We have derived Markov Chain Monte Carlo algorithms for each model variant using a combination of Gibbs sampling and Metropolis-Hastings steps to infer the parameters given observed data. These have been demonstrated on synthetic data, where the correct assumption about the hidden source distribution was shown to give optimal performance, artifically mixed audio data, where the sources were successfully recovered, gene expression data, where the results were consistent with those using standard ICA, and financial data, where we found improved performance over FastICA. A MATLAB implementation of the algorithms will be made available at http://learning.eng.cam.ac.uk/zoubin/.

There are a number of directions in which this work can be extended. Although powerful, the current algorithm is far too slow for real-time applications. Improved simulation speech could be achieved by using more efficient Monte Carlo methods such as Hamiltonian Monte Carlo or overrelaxation. Faster partially deterministic algorithms would be useful for online learning in applications such as audio processing. It is anticipated that the sparse nature of the model will make it appropriate for large datasets. In the case of time series data the ability of model to switch sources on or off will be valuable in applications where sources may be active only for given periods of time, for example in blind source separation problems such as the cocktail party problem, where people are likely to be speaking only for certain periods. In this siutation the extension of the model to use a Hidden Markov Model (HMM) to use the dependence between subsequent source values could improve performance.

References

[1] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems*. 8:757–763. The MIT Press, 1996.

[2] H. Attias. Independent factor analysis. *Neural Computation*, 1999.

[3] A. D. Back and A. S. Weigend. A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural System*, 8:473–484, 1997.

[4] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[5] Jean-Francois Cardoso. High-Order Constrasts for Independent Component Analysis. *Neural Comp.*, 11(1):157–192, 1999.

[6] J.F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, April 1997.

[7] R. Choudrey and S. Roberts. Flexible bayesian independent component analysis for blind source separation. In *3rd International Conference on Independent Component Analysis and Blind Signal Separation*, pages 90–95, 2001.

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[9] Z. Ghahramani, T.L. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. In *Bayesian Statistics 8*. Oxford University Press, 2007.

[10] Gilks, W. R. and Wild, P. Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, 41(2):337–348, 1992.

[11] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. Technical Report 1, Gatsby Computational Neuroscience Unit, 2005.

[12] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[13] N. D. Lawrence and C. M. Bishop. Variational bayesian independent component analysis. Technical report, Computer Laboratory, University of Cambridge, 2000.

[14] D. MacKay. Maximum likelihood and covariant algorithms for independent component analysis, 1996.

[15] D.J.C. MacKay. Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[16] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski. Independent component analysis of electroencephalographic data. *Advances in Neural Information Processing Systems*, 8:145–151, 1996.

[17] Ann-Marie Martoglio, James W. Miskin, Stephen K. Smith, and David J. C. MacKay. A decomposition model to track gene expression signatures: preview on observer-independent classification of ovarian cancer. *Bioinformatics*, 18(12):1617–1624, 2002.

[18] Edward Meeds, Zoubin Ghahramani, Radford Neal, and Sam Roweis. Modeling dyadic data with binary latent factors. In *Neural Information Processing Systems*, volume 19, 2006.

[19] B. A. Pearlmutter and L. C. Parra. A context-sensitive generalization of ica. In *International Conference on Neural Information Processing*, 1996.

[20] Sylvia Richardson and Peter J. Green. On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society*, 59:731–792, 1997.

[21] Y.W. Teh, D. Gorur, and Z. Ghahramani. Stick-breaking construction for the indian buffet. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-2007)*, 2007.

A Deriving conditional distributions

A.1 Hidden sources.

We only wish to sample x_{kt} when $z_{kt} = 1$, since otherwise x_{kt} has no influence on the likelihood function. Since the columns of $\mathbf{Y}$ are generated independently from the corresponding columns of $\mathbf{X}$ we can simplify $P(x_{kt}|\mathbf{G}, \mathbf{X}_{-kt}, \mathbf{Y}, \mathbf{Z}) = P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t)$. Now we can use Baye's rule: $P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) \propto P(\mathbf{y}_t|\mathbf{G}, \mathbf{x}_t, \mathbf{z}_t, \sigma_\epsilon^2)P(x_{kt})$. We denote the k-th column of $\mathbf{G}$ by $\mathbf{g}_k$.

A.1.1 Infinite Sparse FA

We have

$$\log P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) = -\frac{1}{2}x_{kt}^2 - \frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}_{-kt} - \mathbf{g}_k x_{kt})^T(\boldsymbol{\epsilon}_{-kt} - \mathbf{g}_k x_{kt}) + \text{const} \tag{90}$$

$$= -\frac{1}{2}x_{kt}^2 - \frac{1}{2\sigma_\epsilon^2}\left(x_{kt}^2\mathbf{g}_k^T\mathbf{g}_k - 2\mathbf{g}_k^T\boldsymbol{\epsilon}_{-kt}x_{kt} + \boldsymbol{\epsilon}_{-kt}^T\boldsymbol{\epsilon}_{-kt}\right) + \text{const} \tag{91}$$

Comparing coefficients to the canonical form of the Gaussian we find

$$P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) = \mathcal{N}\left(x_{kt}; \frac{\mathbf{g}_k^T\boldsymbol{\epsilon}_{-kt}}{\sigma_\epsilon^2 + \mathbf{g}_k^T\mathbf{g}_k}, \frac{\sigma_\epsilon^2}{\sigma_\epsilon^2 + \mathbf{g}_k^T\mathbf{g}_k}\right) \tag{92}$$

A.1.2 Infinite ICA

The Laplacian prior on x_{kt} is $P(x_{kt}) = \frac{1}{2}\exp(-|x_{kt}|)$. Thus we have

$$\log P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) = -|x_{kt}| - \frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}_{-kt} - \mathbf{g}_k x_{kt})^T(\boldsymbol{\epsilon}_{-kt} - \mathbf{g}_k x_{kt}) + \text{const} \tag{93}$$

$$= -|x_{kt}| - \frac{1}{2\sigma_\epsilon^2}\left(x_{kt}^2\mathbf{g}_k^T\mathbf{g}_k - 2\mathbf{g}_k^T\boldsymbol{\epsilon}_{-kt}x_{kt} + \boldsymbol{\epsilon}_{-kt}^T\boldsymbol{\epsilon}_{-kt}\right) + \text{const} \tag{94}$$

Comparing coefficients to the canonical form of the Gaussian we find the result is a piecewise Gaussian distribution.

$$P(x_{kt}|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) = \begin{cases} \frac{B_+}{A}\mathcal{N}(x_{kt}; \mu_+, \sigma) & x_{kt} > 0 \\ \frac{B_-}{A}\mathcal{N}(x_{kt}; \mu_-, \sigma) & x_{kt} < 0 \end{cases} \tag{95}$$

where

$$\mu_+ = \frac{\mathbf{g}_k^T \boldsymbol{\epsilon}_{-kt} - \sigma_\epsilon^2}{\mathbf{g}_k^T \mathbf{g}_k} \tag{96}$$

$$\mu_- = \frac{\mathbf{g}_k^T \boldsymbol{\epsilon}_{-kt} + \sigma_\epsilon^2}{\mathbf{g}_k^T \mathbf{g}_k} \tag{97}$$

$$\sigma^2 = \frac{\sigma_\epsilon^2}{\mathbf{g}_k^T \mathbf{g}_k} \tag{98}$$

and B_+ and B_- are chosen such that the distribution is continuous and A such that it is correctly normalised.

$$B_+ = \mathcal{N}(0; \mu_-; \sigma) \tag{99}$$

$$B_- = \mathcal{N}(0; \mu_+; \sigma) \tag{100}$$

$$A = A_- B_- + A_+ B_+ \tag{101}$$

where $A_- = F(0; \mu_-, \sigma)$ and $A_+ = 1 - F(0; \mu_+, \sigma)$. To sample from this distribution we need to calculate its cdf and then invert it. Let

$$u(x) = \int_{-\infty}^{x} P(x'|\mathbf{G}, \mathbf{x}_{-kt}, \mathbf{y}_t, \mathbf{z}_t) dx' \tag{102}$$

$$= \begin{cases} \frac{A_- B_-}{A} + \frac{B_+}{A}(F(x; \mu_+, \sigma) - (1 - A_+)) & x_{kt} > 0 \\ \frac{B_-}{A} F(x; \mu_-, \sigma) & x_{kt} < 0 \end{cases} \tag{103}$$

For $u > \frac{A_- B_-}{A}$ we have

$$u = 1 + \frac{B_+}{A}(F(x; \mu_+, \sigma) - 1)$$

$$\Rightarrow x = F^{-1}(\frac{A}{B_+}(u - 1) + 1; \mu_+, \sigma) \tag{104}$$

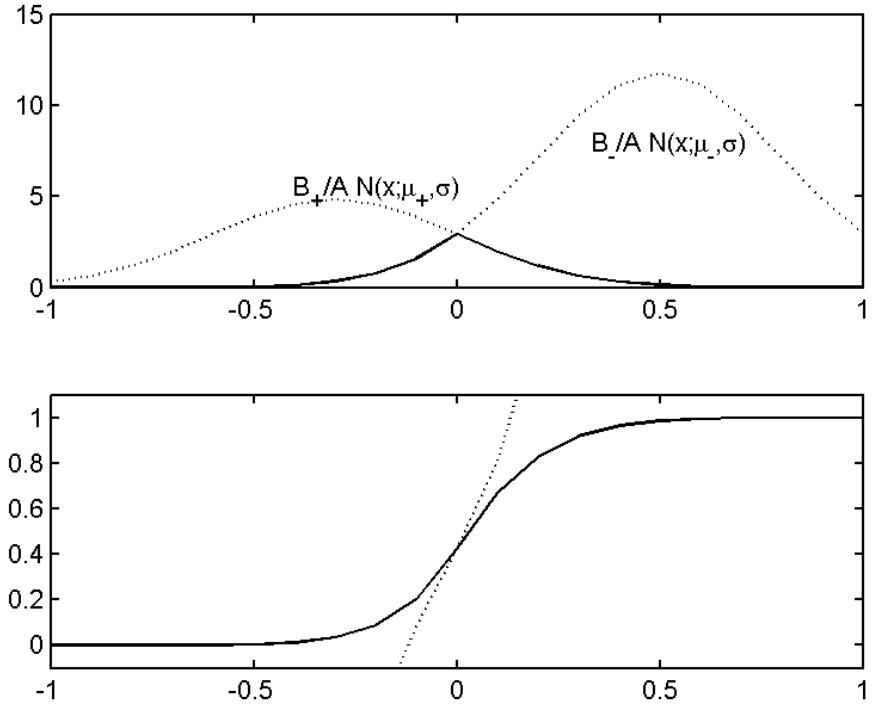


Figure 19: Example piecewise Gaussian of Equation (102) with $\mu_+ = -0.3, \mu_- = 0.5, \sigma = 0.3$. Upper panel shows the pdf (note correct scaling to give continuity). Lower panel shows cdf (note correct normalization). Continuous black line is the piecewise Gaussian, dotted lines denote the component Gaussians.

For $u < \frac{A_- B_-}{A}$ we find

$$x = F^{-1}(\frac{A}{B_-}u; \mu_-, \sigma) \tag{105}$$

Thus to sample x we draw $u \sim$Uniform(0,1): if $u < \frac{A_- B_-}{A}$ we calculate x using Equation 105, otherwise we use Equation 104. An example of this piecewise Gaussian is shown in Figure 19.

A.2 Sampling G

We sample individual columns of the mixing matrix $\mathbf{G}$. We denote the kth column of $\mathbf{G}$ by $\mathbf{g}_k$ and the kth row of $(\mathbf{Z} \circ \mathbf{X})$ by ${\mathbf{s}_k}^T$. The likelihood function

Algorithm 2 Sample from piecewise Gaussian distribution of Equation 96

$A \leftarrow F(0;\mu_+,\sigma) + 1 - F(0;\mu_-,\sigma)$
$u \leftarrow U(0, A)$
if $u < F(0;\mu_+,\sigma)$ **then**
  return $F^{-1}(u;\mu_+,\sigma)$
else
  return $F^{-1}(u+1-A;\mu_-,\sigma)$
end if

$P(\mathbf{Y}|\mathbf{G},\mathbf{X},\mathbf{Z},\sigma_\epsilon^2)$ has exponent $-\frac{1}{2\sigma_\epsilon^2}\times$

$$\begin{aligned}
\operatorname{tr}(\mathbf{E}^T\mathbf{E}) &= \operatorname{tr}((\mathbf{E}|_{\mathbf{g}_k=0} - \mathbf{g}_k{\mathbf{s}_k}^T)^T(\mathbf{E}|_{\mathbf{g}_k=0} - \mathbf{g}_k{\mathbf{s}_k}^T)) && (106)\\
&= \operatorname{tr}(\mathbf{s}_k\mathbf{g}_k^T\mathbf{g}_k{\mathbf{s}_k}^T - 2\mathbf{s}_k\mathbf{g}_k^T\mathbf{E}|_{\mathbf{g}_k=0} + \text{const} && (107)\\
&= x_i g_j g_j x_i - 2x_i g_j(\mathbf{E}|_{\mathbf{g}_k=0})_{ji} + \text{const} && (108)\\
&= ({\mathbf{s}_k}^T\mathbf{s}_k)(\mathbf{g}_k^T\mathbf{g}_k) - 2\mathbf{g}_k^T\mathbf{E}|_{\mathbf{g}_k=0} + \text{const}\mathbf{s}_k && (109)
\end{aligned}$$

using summation notation.

A.3 Sampling σ_ϵ^2

Taking the log of the total likelihood from Equation (49) gives

$$\log P(\mathbf{E}|\sigma_\epsilon^2) = -\frac{1}{2\sigma_\epsilon^2}\operatorname{tr}(\mathbf{E}^T\mathbf{E}) - \frac{ND}{2}\log 2\pi\sigma_\epsilon^2 \tag{110}$$

Similarly taking the log of the prior on σ_ϵ^2 defined in Equation (33), we have

$$\log P(\sigma_\epsilon^2|a,b) = -(a+1)\log\sigma_\epsilon^2 - \frac{1}{b\sigma_\epsilon^2} - \log\Gamma(a) - a\log b \tag{111}$$

Applying Bayes' rule and using the prior from Equation (33) we find

$$P(\sigma_\epsilon^2|\mathbf{E}, a, b) \propto P(\mathbf{E}|\sigma_\epsilon^2)P(\sigma_\epsilon^2|a, b) \tag{112}$$

$$\Rightarrow \log P(\sigma_\epsilon^2|\mathbf{E}) = \log P(\mathbf{E}|\sigma_\epsilon^2) + \log P(\sigma_\epsilon^2|a, b) \tag{113}$$

$$= -[(a+1) + \frac{ND}{2}] \log \sigma_\epsilon^2 - [\frac{1}{b} + \frac{1}{2}\operatorname{tr}(\mathbf{E}^T\mathbf{E})]\frac{1}{\sigma_\epsilon^2} + \text{const.} \tag{114}$$

$$\Rightarrow P(\sigma_\epsilon^2|\mathbf{E}, a, b) = \mathcal{IG}\left(\sigma_\epsilon^2; a + \frac{ND}{2}, \frac{b}{1 + \frac{b}{2}\operatorname{tr}(\mathbf{E}^T\mathbf{E})}\right) \tag{115}$$

by equating coefficients to the canonical form of the inverse Gamma distribution.

A.4 Sampling σ_G^2

For sampling σ_G^2 the conditional prior on $\mathbf{G}$ acts like the likelihood since the likelihood itself is independent of σ_G^2 given $\mathbf{G}$

$$P(\sigma_G^2|\mathbf{G}, c, d) \propto P(\mathbf{G}|\sigma_G^2)P(\sigma_G^2|c, d) \tag{116}$$

$$\Rightarrow \log P(\sigma_G^2|\mathbf{G}, c, d) = -\frac{1}{2\sigma_G^2}\operatorname{tr}(\mathbf{G}^T\mathbf{G}) - \frac{DK}{2}\log 2\pi\sigma_G^2 - (c+1)\log \sigma_G^2 - \frac{1}{\sigma_G^2 d} \tag{117}$$

$$\Rightarrow P(\sigma_G^2|\mathbf{G}, c, d) = \mathcal{IG}\left(\sigma_G^2; c + \frac{DK}{2}, \frac{d}{1 + \frac{d}{2}\operatorname{tr}(\mathbf{G}^T\mathbf{G})}\right) \tag{118}$$

A.5 Sampling α

For sampling α the conditional prior on $\mathbf{Z}$, given by Equation (44), acts as the likelihood since the likelihood itself is independent of α given $\mathbf{Z}$. Taking the log of this expression gives

$$\log P(\mathbf{Z}|\alpha) = K_+ \log \alpha - H_N(\beta)\alpha + \text{const.} \tag{119}$$

Using the Gamma prior on α with Baye's rule we have

$$P(\alpha|\mathbf{G}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \beta) = P(\alpha|\mathbf{Z}, \beta) \propto P(\mathbf{Z}|\alpha, \beta)P(\alpha) \tag{120}$$

$$\Rightarrow \log P(\alpha|\mathbf{Z}, \beta) = (K_+ + e - 1) \log \alpha - \left(H_N(\beta) + \frac{1}{f} \right) \alpha + \text{const.} \tag{121}$$

$$\Rightarrow P(\alpha|\mathbf{Z}) = \mathcal{G}\left(\alpha; K_+ + e, \frac{f}{1 + fH(\beta)} \right) \tag{122}$$

B Financial data

B.1 Raw data

B.2 Transformed data

C Main algorithm

Algorithm 3 MCMC sampler for infinite ICA

initialise $\mathbf{G}, \mathbf{X}, \mathbf{Z}$ using their priors
for $r = 1 \cdots$ number of iterations **do**
  for $t = 1 \cdots N$ **do**
    for $k = 1 \cdots K$ **do**
      if $m_{k,-t} > 0$ **then**
        sample z_{kt} according to Equation (56)
        if $z_{kt} = 1$ **then**
          sample x_{kt} according to Section 3.2
        end if
      else
        mark z_{kt} to be zeroed
      end if
    end for
    zero marked z_{kt}'s
    sample κ_t according to Section 3.4
    $\mathbf{Z}_{K+1:K+\kappa_t,t} \leftarrow 1$
    initialise $\mathbf{G}_{:,K+1:K+\kappa_t}$ from prior
    for all $x_{kt} \in \mathbf{X}_{K+1:K+\kappa_t,t}$ **do**
      sample x_{kt} according to Section 3.2
    end for
    $K \leftarrow K + \kappa_t$
  end for
  for $k = 1 \cdots K$ **do**
    sample $\mathbf{g}_k$ according to Equations (67) and (68)
  end for
  remove rows with $m_k = 0$ from $\mathbf{Z}$
  remove corresponding rows from $\mathbf{X}$ and columns from $\mathbf{G}$
  sample $\sigma_\epsilon^2, \sigma_G^2, \alpha$ from Equations (115), (70), (71) respectively
end for