Bayesian Sparse Unsupervised Learning for Probit Models of Binary Data

Ari Pakman
ari@stat.columbia.edu

Ben Shababo
bms2156@columbia.edu

Liam Paninski
liam@stat.columbia.edu

Department of Statistics, Center for Theoretical Neuroscience
and Grossman Center for the Statistics of Mind,
Columbia University, New York, NY 10027

Abstract

We present a unified approach to unsupervised Bayesian learning of factor models for binary data with binary and spike-and-slab latent factors. We introduce a non-negative constraint in the spike-and-slab prior that eliminates the usual sign ambiguity present in factor models and lowers the generalization error on the datasets tested here. For the generative models we use probit functions, which can be sampled without tuning parameters, unlike previous works that used logistic functions. The posterior distributions involve mixtures of binary and truncated Gaussian variables, for which we present exact Hamiltonian Monte Carlo samplers and compare their properties to Gibbs samplers.

1 INTRODUCTION

Several studies in recent years have shown that factor models for unsupervised learning of partially observed data can successfully avoid overfitting by exploiting two key elements. The first is a Bayesian approach that samples from the model parameters instead of committing to a single MAP estimate [Salakhutdinov and Mnih, 2008, Mohamed et al., 2008]. The second element is the use of sparse or discrete latent variables, which also contributes to the interpretability of the generative models.

When the latent variables are continuous, the most popular approach to achieve sparsity in a MAP setting is through the L_1 penalty [Olshausen and Field, 1997]. But recent results suggest that a Bayesian approach with a spike-and-slab prior can lead to better performance [Mohamed et al., 2012], both computationally and in terms of predictive ability, since Bayesian models can avoid the costly tuning of the sparsity hyperparameters via cross-validation.

An alternative form of latent sparsity occurs when the latent variables are binary instead of continuous and such models have also shown very good performance in many cases.

In this paper we focus on binary data and present a unified Bayesian approach to explore, in a given dataset, which of several forms of latent sparsity best describes the observations. Along with binary and spike-and-slab latent variables, we introduce a spike-and-slab model with a non-negativity constraint on the hidden factors. This is a form of non-negative matrix factorization that lowers the generalization error of the spike-and-slab model on the datasets tested here and, to our knowledge, has not been studied before. Moreover, by eliminating the usual sign ambiguity present in factor models we obtain, as for binary latent variables, a natural interpretation of the signs of the learned factor loadings. An important motivation for this new model is the analysis of spiking activity of neural populations, where the signs of the factor loadings indicate the excitatory or inhibitory nature of the latent inputs.[1]

Another aspect of this work is the use of probit functions in the generative models. Compared to logistic functions, these probit models lead to posterior distributions that can be sampled efficiently without tuning of sampling parameters. We also present some empirical evidence that indicates a faster mixing rate than logistic based models. Exploiting the latent variable interpretation of the probit function, Monte Carlo posterior inference requires sampling from mixtures of binary and truncated Gaussian variables. We present the details of exact Hamiltonian Monte Carlo (HMC) samplers for these distributions, based on the recent results of [Pakman and Paninski, 2013a,b], and compare them to simpler Gibbs samplers.

[1] We are currently pursuing applications to large neural datasets; the results will appear elsewhere.

2 GENERATIVE MODELS

The observations are i.i.d. binary vectors

$$\mathbf{x}_t \in \{-1,+1\}^N \qquad t = 1,\ldots,T$$

which we model using K latent sparse factors. In general, we consider the case where only a subset of the entries in each $\mathbf{x}_t$ are observed. This allows a more general application of our method as well as a means for evaluating the quality of the learned model.

We will consider three generative models, which differ in the structure of the latent factors:

Model 1: Binary Factors

Each $\mathbf{x}_t = (x_{1t},\ldots,x_{Nt})$ is generated as

$$\begin{aligned} s_{kt}|a_k &\sim \begin{cases} 1 & \text{with prob. } a_k\,, \\ 0 & \text{with prob. } 1-a_k\,, \end{cases} \\ x_{nt}|\mathbf{z}_n,\mathbf{s}_t &\sim \begin{cases} +1 & \text{with prob. } \Phi\left(\mathbf{z}_n\cdot\mathbf{s}_t\right)\,, \\ -1 & \text{with prob. } 1-\Phi\left(\mathbf{z}_n\cdot\mathbf{s}_t\right)\,, \end{cases} \end{aligned} \tag{1}$$

where

$$\mathbf{z}_n\cdot\mathbf{s}_t = \sum_{k=1}^{K+1} z_{nk}s_{kt} \qquad n = 1,\ldots,N\,, \tag{2}$$

$z_{n(K+1)}$ is a constant offset, and we defined $s_{(K+1)t}\equiv 1$. In this model the latent factors s_{kt} are binary variables and the values of the observations x_{nt} are sampled from a Bernoulli distribution with parameter

$$\Phi(q) = \frac{1}{\sqrt{2\pi}}\int_0^\infty dy\; e^{-\frac{1}{2}(y-q)^2} \tag{3}$$

which is the probit function. Using

$$\Phi(q) + \Phi(-q) = 1\,,$$

we can express (1) as

$$p(x_{nt}|\mathbf{s}_t,\mathbf{z}_n) = \Phi(x_{nt}\mathbf{z}_n\cdot\mathbf{s}_t)\,.$$

Model 2: Spike-and-Slab Factors

In this model, we add an additional layer after the binary variables s_{kt}. Each $\mathbf{x}_t$ is now generated as

$$\begin{aligned} s_{kt}|a_k &\sim \begin{cases} 1 & \text{with prob. } a_k\,, \\ 0 & \text{with prob. } 1-a_k\,, \end{cases} && (4) \\ v_{kt}|s_{kt} &\sim \mathcal{N}(0,1) && (5) \\ x_{nt}|\mathbf{z}_n,\mathbf{f}_t &\sim \begin{cases} +1 & \text{with prob. } \Phi\left(\mathbf{z}_n\cdot\mathbf{f}_t\right)\,, \\ -1 & \text{with prob. } 1-\Phi\left(\mathbf{z}_n\cdot\mathbf{f}_t\right)\,, \end{cases} \end{aligned}$$

where

$$f_{kt} \;=\; s_{kt}v_{kt}\,,$$

and $\mathbf{z}_n\cdot\mathbf{f}_t$ is defined as in (2), with $f_{(K+1)t}\equiv 1$. From (4) and (5) it follows that the latent factors are sampled from

$$f_{kt} \sim a_k\mathcal{N}(0,1) + (1-a_k)\delta(f_{kt}) \qquad k = 1,\ldots,K \tag{6}$$

which is a spike-and-slab distribution [George and McCulloch, 1993, Mitchell and Beauchamp, 1988]. Note that this distribution achieves sparsity by assigning a positive probability mass to the event $f_{kt} = 0$.

Model 3: Non-negative Spike-and-Slab Factors

This model is equal to Model 2, but we truncate the slab variables in (5) to be $v_{kt}\geq 0$. In this case and in Model 1, the non-negativity of f_{kt} or s_{kt} allows us to interpret their influence on x_{nt} as increasing its rate ($z_{nk} > 0$) or decreasing its rate ($z_{nk} < 0$). As mentioned above, this sign is relevant in the analysis of neural populations, where the signs of z_{nk} indicate the excitatory or inhibitory nature of the latent inputs.

2.1 Prior on the factor loadings

We use the spike-and-slab prior to regularize the learned values of z_{nk},

$$\begin{aligned} c_{nk}|b &\sim \begin{cases} 1 & \text{with prob. } b\,, \\ 0 & \text{with prob. } 1-b\,, \end{cases} \\ z_{nk}|c_{nk} &\sim \begin{cases} \delta(z_{nk}) & \text{for } \; c_{nk} = 0\,, \\ \mathcal{N}(0,\tau^2) & \text{for } \; c_{nk} = 1\,. \end{cases} \end{aligned}$$

One should distinguish the sparseness of $\mathbf{s}_t$ or $\mathbf{f}_t$, which is inherent to the data generating process, from this regularizing sparsity prior on z_{nk}. One could also add non-negativity constraints on z_{nk}, although we have not explored this possibility.

2.2 Hyperparamters

For the hyperparameters a_k, b and τ^2 we will assume flat priors and sample from the posterior, although one can also consider Beta priors on a_k, b, and an inverse-Gamma prior on τ^2.

2.3 Relation to previous works

Model 1 is the simplest example of a sigmoid belief network [Neal, 1990, 1992], having only one layer of hidden units. It is similar to a restricted Boltzmann machine [Smolensky, 1986, Hinton, 2002], but with *directed* connections from the hidden to the visible units.

Model 2 is similar to the spike-and-slab generative model for binary data studied in [Mohamed et al., 2012]. Bayesian factor models for binary data with

non-sparse latent variables were studied in [Mohamed et al., 2008].

An important aspect in which our Models 1 and 2 differ from [Neal, 1990, 1992, Mohamed et al., 2012] is that those works used logistic sigmoid functions, while we use probit functions. So although the generative models are similar, the inference algorithms are quite different, as we discuss next.

An alternative two-layer directed Bayesian network for binary variables, with a different generative structure than Model 1, is the Noisy-Or network, which is popular in the field of medical diagnostics [Shwe et al., 1991]. For other approaches to factor models of binary data see [Zhou et al., 2012, Pillow and Scott, 2012].

3 INFERENCE

As we mentioned above, in general we will not observe all the $N \times T$ elements of $\mathbf{X}$, but only a subset. Let us denote by $\mathcal{O}$ the indices of the observed subset, and by $\mathcal{O}_n$ and $\mathcal{O}_t$ the indices of the observed entries in the rows $\mathbf{x}_n$ and columns $\mathbf{x}_t$ of $\mathbf{X}$, respectively. By an abuse of notation, in the posterior distributions below we use $\mathbf{X}$ to denote the subset $\mathcal{O}$ of observed entries of $\mathbf{X}$.

3.1 Model 1

As usual with probit models, we consider (3) as a marginal over a non-negative random variable y. Given a set of observations x_{nt}, we are interested in sampling from the posterior distribution

$$p(\mathbf{S}, \mathbf{Y}, \mathbf{Z}, \mathbf{c}, \mathbf{a}, b, \tau^2 | \mathbf{X}) \propto \prod_{(n,t)\in\mathcal{O}} p(x_{nt}, y_{nt} | \mathbf{s}_t, \mathbf{z}_n) \times \prod_t p(\mathbf{s}_t|\mathbf{a}) \prod_n p(\mathbf{z}_n|\mathbf{c}_n, \tau^2) p(\mathbf{c}_n|b) \quad (7)$$

where we defined

$$p(x_{nt}, y_{nt}|\mathbf{s}_t, \mathbf{z}_n) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y_{nt} - x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t)^2}$$

with $y_{nt} \geq 0$, and

$$\begin{aligned} p(\mathbf{s}_t|\mathbf{a}) &= \prod_{k=1}^{K} a_k^{s_{kt}} (1-a_k)^{1-s_{kt}} \\ p(\mathbf{z}_n|\mathbf{c}_n, \tau^2) &= \prod_{k=1}^{K+1} p(z_{nk}|c_{nk}, \tau^2) \\ p(\mathbf{c}_n|b) &= \prod_{k=1}^{K+1} b^{c_{nk}} (1-b)^{1-c_{nk}} \end{aligned}$$

We can Gibbs sample in blocks from (7) by iterating over:

$$\prod_t p(\mathbf{s}_t, \mathbf{y}_t | \mathbf{x}_t, \mathbf{Z}, \mathbf{a}) \quad (8)$$

$$\prod_n p(\mathbf{z}_n, \mathbf{c}_n | \mathbf{x}_n, \mathbf{y}_n, \mathbf{S}, \tau^2, b) \quad (9)$$

$$p(a_k|\mathbf{s}_k) = \text{Beta}\left(|\mathbf{s}_{k+}|, |\mathbf{s}_{k-}|\right)$$

$$p(b|\mathbf{c}) = \text{Beta}\left(|\mathbf{c}_+|, |\mathbf{c}_-|\right)$$

$$p(\tau^2|\mathbf{Z}) = \text{InvGamma}\left(\frac{|\mathbf{c}_+|}{2} + 1, \frac{\|\mathbf{Z}\|^2}{2}\right).$$

For the binary vectors $\mathbf{c}$, the notation $|\mathbf{c}_+|$ and $|\mathbf{c}_-|$ indicates the total number of components with $c_{nk} = 1$ and $c_{nk} = 0$, respectively, for all n, k. A similar notation is used for $\mathbf{s}_k$. Note the factorized form of the distributions (8) and (9), which allows us to parallelize the sampling over t and n, respectively.

For sampling from the distribution over the factor loadings (9), we present the details in Appendix A.

3.2 Models 2 and 3

In these cases we are interested in sampling from the distribution

$$p(\mathbf{S}, \mathbf{V}, \mathbf{Y}, \mathbf{Z}, \mathbf{c}, \mathbf{a}, b, \tau^2 | \mathbf{X}) \propto \prod_{(n,t)\in\mathcal{O}} p(x_{nt}, y_{nt}|\mathbf{f}_t, \mathbf{z}_n) \prod_t p(\mathbf{s}_t|\mathbf{a}) p(\mathbf{v}_t) \prod_n p(\mathbf{z}_n|\mathbf{c}_n, \tau^2) p(\mathbf{c}_n|b). \quad (10)$$

The steps of the Gibbs sampler are similar to Model 1, but instead of (8) we now sample from

$$\prod_t p(\mathbf{y}_t, \mathbf{v}_t, \mathbf{s}_t | \mathbf{x}_t, \mathbf{Z}, \mathbf{a}). \quad (11)$$

3.3 Exact HMC vs. Gibbs sampler

For each t, the distribution (8) is

$$p(\mathbf{s}_t, \mathbf{y}_t | \mathbf{x}_t, \mathbf{Z}, \mathbf{a}) \propto \prod_{n\in\mathcal{O}_t} e^{-\frac{1}{2}(y_{nt} - x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t)^2} \prod_k a_k^{s_{kt}} (1-a_k)^{1-s_{kt}} \quad (12)$$

with $y_{nt} \geq 0$. For this class of mixtures of binary and truncated Gaussian variables, exact HMC techniques have been developed recently [Pakman and Paninski, 2013b]. In appendix B.1 we present the details of the exact HMC algorithm applied to (12).[2] But for the particular case of (12) we expect the simpler Gibbs sampler to perform well, because when alternating between $p(\mathbf{s}_t|\mathbf{y}_t, \mathbf{x}_t, \mathbf{Z}, \mathbf{a})$ and

$$p(\mathbf{y}_t|\mathbf{s}_t, \mathbf{x}_t, \mathbf{Z}, \mathbf{a}) \propto \prod_{n\in\mathcal{O}_t} e^{-\frac{1}{2}(y_{nt} - x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t)^2}, \quad (13)$$

[2] The *joint* distribution $p(\mathbf{S}, \mathbf{Y}, \mathbf{Z}, \mathbf{c}|\mathbf{X}, \mathbf{a}, b, \tau^2)$ can also be sampled using exact HMC but it is preferable to split the sampling into the two steps (8) and (9) in order to parallelize over t and n, respectively.

the factorization over n in the above expression leads to a very fast mixing of the Markov chain. Although we have not performed a thorough comparison, our preliminary findings confirm that the Gibbs and HMC samplers indeed have comparable mixing times when sampling from (12).

The situation is different for the distributions in (11),

$$p(\mathbf{y}_t, \mathbf{v}_t, \mathbf{s}_t | \mathbf{x}_t, \mathbf{Z}, \mathbf{a}) \tag{14}$$
$$\propto \prod_{n \epsilon \mathcal{O}_t} e^{-\frac{1}{2}(y_{nt} - x_{nt}\mathbf{z}_n \cdot \mathbf{f}_t)^2} e^{-\frac{|\mathbf{v}_t|^2}{2}} \prod_k a_k^{s_{kt}} (1-a_k)^{1-s_{kt}},$$

in particular for Model 3, which restricts to $y_{nt}, v_{kt} \geq 0$. In these truncated cases, with correlations between the v_{kt} variables, the exact HMC approach is known to mix much faster than Gibbs [Pakman and Paninski, 2013a]. In Section B.2 we present the details of the exact HMC sampler for this distribution.

It is worth stressing that when using probit functions there are no parameters to fine tune in the samplers. The only free parameter in the exact HMC algorithm is the total travel time τ_{max}. In examples we present below we used $\tau_{max} = 23$, but the results were independent of τ_{max} in a wide range of values. Note that in the logistic model of [Mohamed et al., 2012] a leapfrog HMC sampler was used that requires parameter tuning to work efficiently, though more recent techniques such as Bayesian optimization [Wang et al., 2013] or NUTS [Hoffman and Gelman, 2011] may be effective here.

4 PREDICTIONS AND MODEL SELECTION

Once we have samples from the posterior distribution, we can estimate the predicted posterior probability for any x_{nt} in Model 1 as

$$p(x_{nt} = 1 | \mathbf{X}) \simeq \frac{1}{M} \sum_{m=1}^{M} \Phi(\mathbf{z}_n^{(m)} \cdot \mathbf{s}_t^{(m)})$$

where $\mathbf{z}_n^{(m)}, \mathbf{s}_t^{(m)}$ are Monte Carlo samples. For Models 2 and 3, we replace $\mathbf{s}_t$ with $\mathbf{f}_t$. From the posterior probabilities we obtain the posterior expected value of x_{nt} as

$$\langle x_{nt} \rangle = 2p(x_{nt} = 1 | \mathbf{X}) - 1 .$$

In general we will be interested in computing these predictions for indices (n,t) in a held-out dataset $\mathcal{T}$ to measure the predictive power of each model or to select the number of factors K. For this we consider

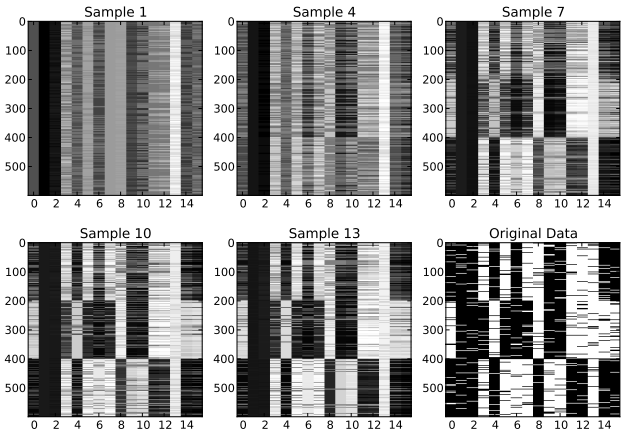

Figure 1: Synthetic dataset. Values of $\Phi(\mathbf{z}_n^{(m)} \cdot \mathbf{s}_t^{(m)})$ at several iterations m in Model 1 with $K = 5$, initialized with random values, and original data in the synthetic model described in the text. Note that the mixing is quite fast and after only 10 iterations the samples resemble the original data.

two measures of performance. The RMSE is given by[3]

$$RMSE^2 = \frac{1}{4|\mathcal{T}|} \sum_{n,t \epsilon \mathcal{T}} (x_{nt} - \langle x_{nt} \rangle)^2$$

and the mean negative log-predictive probability is

$$MNLP = -\frac{1}{|\mathcal{T}|} \sum_{n,t \epsilon \mathcal{T}} \log_2 p(x_{nt} | \mathbf{X}) .$$

Note that a blind model which assigns probability 1/2 to both $x = \pm 1$ gives $MNLP = 1$, and as the quality of the predictive model improves we get smaller MNLP values. If all we learn from the training dataset is the proportion $\hat{p}$ of data with $x = 1$, the learned model assigns probability $\hat{p}$ to $x = 1$ and yields

$$MNLP_{\hat{p}} = -\hat{p} \log_2(\hat{p}) - (1-\hat{p}) \log_2(1-\hat{p}) .$$

5 EXPERIMENTS

We implemented the algorithm in Python, using CUDA in a GPU for the HMC sampler in order to take advantage of the parallelization over t. We show results for one synthetic and two real datasets.

The synthetic dataset is similar to that described in [Mohamed et al., 2008]. Three 16-dimensional binary vectors were created with entries randomly set to ± 1 with equal probability. Each vector was repeated 200 times, giving $T = 600$. Finally, each of the 600×16 binary variables is flipped with probability .1. Figure 1

[3]The factor of 1/4 normalizes the RMSE to correspond to a binary coding of 0/1 instead of –1/1.

shows the values of the 600×16 matrix $\Phi(\mathbf{z}_n^{(m)} \cdot \mathbf{s}_t^{(m)})$ at different iterations of a MCMC chain using Model 1 with $K = 5$ and initialized with random values. Note that after 10 iterations the samples closely resemble the original data. Although a more thorough exploration of this point is needed, the probit model seems to mix faster than results reported for logistic functions. For example, in a similar dataset in Figure 2.3 of [Mohamed, 2012] it is shown that for a generative model with a logistic function (without sparse latent variables), the samples obtained using leapfrog HMC only resemble the original data after about 50 MCMC iterations.

Figures 2, 3, and 4 show MNLP values for the three datasets. We omit plots of RMSE values because they are qualitatively similar. The **animals attributes** dataset from [Kemp and Tenenbaum, 2008] consists of $T = 33$ animal species and $N = 102$ ecological and biological properties that are present or absent for each species. Note that this is a case of '$p > n$' dataset. The **SPECT** dataset, available at [Bache and Lichman, 2013], is from [Kurgan et al., 2001] and consists of data from cardiac tomography images from $T = 267$ patients. The $N = 23$ variables consist of 22 binary feature patterns extracted from the images plus an indicator that classifies each patient as normal/abnormal.

We randomly split the datasets into training and test subsets with 90% and 10% of the data respectively. We repeated this split fives times and run the sampler in each case. The figures show the mean and standard deviation over the five cases. We used 120 MCMC iterations and discarded the first 30 iterations as burn-in.

The figures illustrate a property observed in [Mohamed et al., 2012] for similar Bayesian models: the overfit as K grows is relatively slow. As is clear from the figures, the different datasets have structures better captured by discrete (Model 1) or continuous (Models 2 & 3) latent variables. For the latter, note that the additional degrees of freedom provided by the continuous latent variables can be detrimental, as is clear in Figures 2 and 4.

Note also the importance of the non-negativity restriction in Model 3, which in all the cases lowers the train and test errors with respect to Model 2.

6 CONCLUSIONS

Our results illustrate clearly that we can improve the predictive ability of generative models with spike-and-slab latent variables by adopting a simpler model with binary latent variables or by imposing nonnegativity

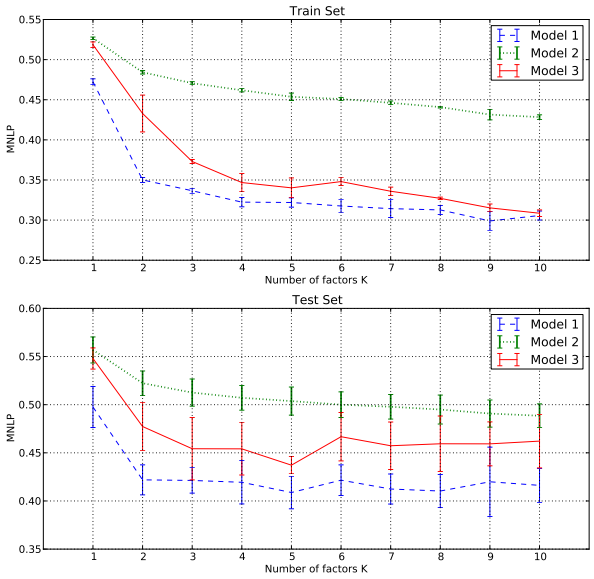

Figure 2: Synthetic dataset. Mean and standard deviations of the MNLP in the train and test for the synthetic model described in the text, as a function of the number K of hidden factors. Note that the models are slow to overfit. The randomly selected test sets give $\text{MNLP}_{\hat{p}} = 0.999$.

constraints on the factors, similar to the case in non-negative matrix factorization of real-valued data. We have also shown that the use of probit functions leads to efficient posterior sampling. Combined with the fact that there is no need to tune sampling parameters in these models, this makes this class of Bayesian probit models an attractive option for factor analysis of binary data.

A SAMPLING THE FACTOR LOADINGS

With a spike-and-slab prior for the factor loadings $\mathbf{z}_n$, we need to sample from the distributions (9)

$$p(\mathbf{z}_n, \mathbf{c}_n | \mathbf{x}_n, \mathbf{y}_n, \mathbf{S}, \tau^2, b) \propto e^{-\frac{1}{2}\mathbf{z}'_{n+}\mathbf{Q}_{n,+}\mathbf{z}_{n+} + \mathbf{j}_{n+}\cdot\mathbf{z}_{n+}} \times \frac{e^{-\frac{\mathbf{z}_{n+}\cdot\mathbf{z}_{n+}}{2\tau^2}}}{(2\pi\tau^2)^{|\mathbf{c}_{n+}|/2}} \delta(\mathbf{z}_{n-})\, b^{|\mathbf{c}_{n+}|}(1-b)^{|\mathbf{c}_{n-}|}. \quad (15)$$

The vector $\mathbf{j}_n$ has components

$$j_{nk} = \sum_{t \in \mathcal{O}_n} x_{nt} s_{kt} y_{nt}, \qquad k = 1, \ldots, K+1$$

and the matrix $\mathbf{Q}_n$ is

$$\mathbf{Q}_n = \sum_{t \in \mathcal{O}_n} \mathbf{s}_t \mathbf{s}_t^T \quad \in \mathbb{R}^{K+1 \times K+1}.$$

In the following we eliminate the subindex n to simplify the notation. We can sample from (15) by decomposing

$$p(\mathbf{z}, \mathbf{c} | \cdots) = p(\mathbf{z} | \mathbf{c}, \cdots) p(\mathbf{c} | \cdots)$$

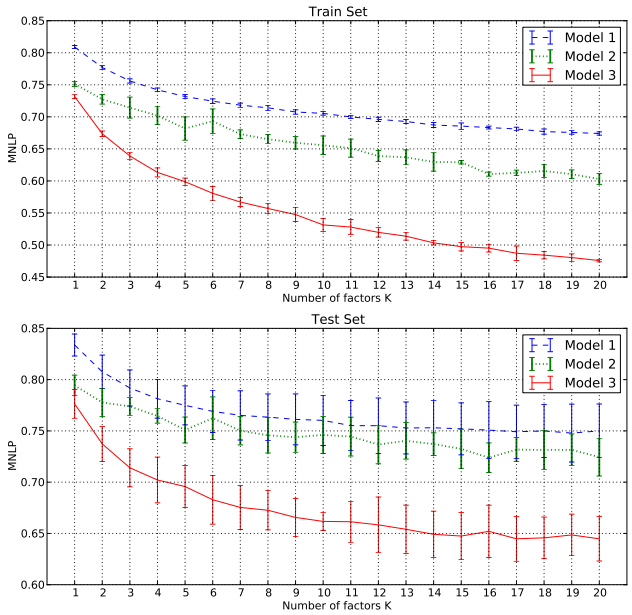

Figure 3: Animals attributes dataset. See the text for a description of the dataset. Note that the positivity restriction of Model 3 leads to a much better performance than Model 2. The randomly selected test sets give $\text{MNLP}_{\hat{p}} = 0.887$.

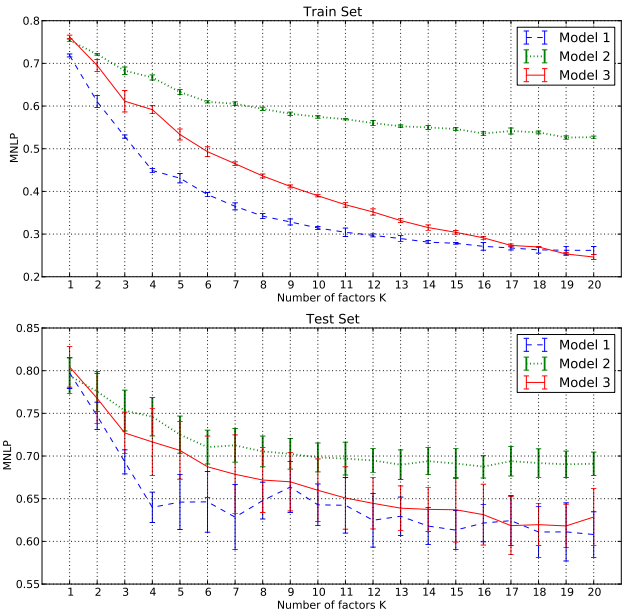

Figure 4: SPECT dataset. See the text for a description of the dataset. Note that the additional degrees of freedom provided by the continuous latent variables in Models 2 and 3 are detrimental to learning the data. The randomly selected test sets give $\text{MNLP}_{\hat{p}} = 0.918$.

where

$$p(\mathbf{z}|\mathbf{c},\cdots) = \mathcal{N}(\mathbf{z}_+|\boldsymbol{\mu}_+,\Sigma_+)\delta(\mathbf{z}_-)$$

$$p(\mathbf{c}|\cdots) \propto \frac{b^{|\mathbf{c}_+|}(1-b)^{|\mathbf{c}_-|}}{\tau^{|\mathbf{c}_+|}} \frac{e^{\frac{\mathbf{j}_+'\boldsymbol{\Sigma}_+\mathbf{j}_+}{2}}}{|\boldsymbol{\Sigma}_+^{-1}|^{1/2}} \tag{16}$$

and we defined

$$\begin{aligned} \boldsymbol{\Sigma}_+^{-1} &= \mathbf{Q}_+ + \mathbb{I}\tau^{-2} \qquad \in \mathbb{R}^{|\mathbf{c}_+|\times|\mathbf{c}_+|} \\ \boldsymbol{\mu}_+ &= \boldsymbol{\Sigma}_+\mathbf{j}_+. \end{aligned}$$

In order to sample the binary vector $\mathbf{c}$ from (16) using a Gibbs or Metropolis sampler, we need the ratios

$$\frac{p(c_k=1|\mathbf{c}_{-k},\cdots)}{p(c_k=0|\mathbf{c}_{-k},\cdots)} = \frac{b}{1-b}\frac{\gamma_k e^{\frac{\gamma_k^2 g_k^2}{2}}}{\tau},$$

where $\mathbf{c}_{-k}$ means that we exclude the kth component, and we defined

$$\begin{aligned} \gamma_k^{-2} &= Q_{kk} + \frac{1}{\tau^2} - \mathbf{q}_{\mathbf{c}_{-k}}^T \boldsymbol{\Sigma}_{\mathbf{c}_{-k}} \mathbf{q}_{\mathbf{c}_{-k}} \\ g_k &= j_k - \mathbf{q}_{\mathbf{c}_{-k}}^T \boldsymbol{\Sigma}_{\mathbf{c}_{-k}} \mathbf{j}_{\mathbf{c}_{-k}}. \end{aligned}$$

Here the notation $\boldsymbol{\Sigma}_{\mathbf{c}_{-k}}$ means that we restrict the indices to those indicated by $\mathbf{c}_{-k}$, and $\mathbf{q}_{\mathbf{c}_{-k}}$ is the k-th row of $\mathbf{Q}$, restricted to the components indicated by $\mathbf{c}_{-k}$.

B EXACT HMC SAMPLING

In this appendix we present the details of the exact HMC sampling algorithms using the techniques developed in [Pakman and Paninski, 2013a,b]

B.1 Model 1

For each t, we are interested in sampling from the mixed binary-continuous distribution (8)

$$\begin{aligned} p(\mathbf{s}_t,\mathbf{y}_t|\mathbf{x}_t,\mathbf{Z},\mathbf{a}) &\propto p(\mathbf{x}_t,\mathbf{y}_t|\mathbf{s}_t,\mathbf{Z})p(\mathbf{s}_t|\mathbf{a}) \qquad (17) \\ &\propto \prod_{n\in\mathcal{O}_t} e^{-\frac{1}{2}(y_{nt}-x_{nt}\mathbf{z}_n\cdot\mathbf{s}_t)^2} \prod_k a_k^{s_{kt}}(1-a_k)^{1-s_{kt}} \end{aligned}$$

with $y_{nt} \geq 0$. The idea is to map this distribution into a piecewise Gaussian by augmenting the binary variables $\mathbf{s}_t$ with continuous vaiables $\mathbf{d}_t$ through the conditional distribution

$$p(\mathbf{d}_t|\mathbf{s}_t) \propto \begin{cases} e^{-\frac{\mathbf{d}_t\cdot\mathbf{d}_t}{2}} & \text{if } s_{kt} = \frac{sign(d_{kt})+1}{2}\ \forall k \\ 0 & \text{otherwise}. \end{cases}$$

Marginalizing over $\mathbf{s}_t$ we get

$$\begin{aligned} p(\mathbf{d}_t,\mathbf{y}_t|\mathbf{x}_t,\mathbf{Z},\mathbf{a}) &= \sum_{\mathbf{s}_t'} p(\mathbf{s}_t',\mathbf{y}_t|\mathbf{x}_t,\mathbf{Z},\mathbf{a})p(\mathbf{d}_t|\mathbf{s}_t') \\ &= p(\mathbf{s}_t,\mathbf{y}_t|\mathbf{x}_t,\mathbf{Z},\mathbf{a})p(\mathbf{d}_t|\mathbf{s}_t)\,. \end{aligned}$$

Note that only one term survives in the sum, which gives

$$\begin{aligned} U_1 &= -\log p(\mathbf{d}_t, \mathbf{y}_t | \mathbf{x}_t, \mathbf{Z}, \mathbf{a}) \qquad (18) \\ &= \frac{1}{2}\sum_n (y_{nt} - x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t)^2 + \frac{\mathbf{d}_t \cdot \mathbf{d}_t}{2} + f(\mathbf{s}_t) + const \end{aligned}$$

where we defined

$$f(\mathbf{s}_t) = -\sum_k s_{kt}\log(a_k) + (1 - s_{kt})\log(1 - a_k). \qquad (19)$$

A sample of the continuous pair $(\mathbf{d}_t, \mathbf{y}_t)$ from (18) gives a binary-continuous sample $(\mathbf{s}_t, \mathbf{y}_t)$ from the original distribution (17) using the simple rule

$$s_{kt} = \frac{sign(d_{kt}) + 1}{2}. \qquad (20)$$

Since (18) is piecewise quadratic, we can sample from it using the exact HMC method. For this, we introduce momentum variables $\mathbf{q}_y$ and $\mathbf{q}_d$ and consider the Hamiltonian

$$H = U_1 + \frac{\mathbf{q}_y \cdot \mathbf{q}_y}{2} + \frac{\mathbf{q}_d \cdot \mathbf{q}_d}{2}.$$

In each iteration we sample initial momenta from standard Normal distributions[4] and let the particle move during a time τ_{max} according to the Hamiltonian equations of motion

$$\dot{\mathbf{y}}_t(\tau) = \frac{\partial H}{\partial \mathbf{q}_y(\tau)}, \qquad \dot{\mathbf{q}}_y(\tau) = -\frac{\partial H}{\partial \mathbf{y}_t(\tau)}, \qquad (21)$$

and similarly for $\mathbf{d}_t$. The final positions of $(\mathbf{d}_t, \mathbf{y}_t)$ belong to a Markov chain with invariant distribution (18).

The solution of (21) for y_{nt} is

$$\begin{aligned} y_{nt}(\tau) &= x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t + \rho_{nt}\cos(\tau) + \dot{y}_{nt}(0)\sin(\tau) \\ &= x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t + u_{y,nt}\sin(\phi_{y,nt} + \tau) \qquad (22) \end{aligned}$$

with

$$\begin{aligned} \rho_{nt} &= y_{nt}(0) - x_{nt}\mathbf{z}_n \cdot \mathbf{s}_t \\ u_{y,nt} &= \sqrt{\rho_{nt}^2 + \dot{y}_{nt}^2(0)} \\ \phi_{y,nt} &= \tan^{-1}(\rho_{nt}/\dot{y}_{nt}(0)) \end{aligned}$$

while the solution for d_{kt} is

$$\begin{aligned} d_{kt}(\tau) &= d_{kt}(0)\cos(\tau) + \dot{d}_{kt}(0)\sin(\tau) \\ &= u_{d,kt}\sin(\phi_{d,kt} + \tau) \qquad (23) \end{aligned}$$

with

$$\begin{aligned} u_{d,kt} &= \sqrt{d_{kt}^2(0) + \dot{d}_{kt}^2(0)} \\ \phi_{d,kt} &= \tan^{-1}(d_{kt}(0)/\dot{d}_{kt}(0)). \end{aligned}$$

[4]The momenta are equal to the velocities in this case, $\dot{\mathbf{y}} = \mathbf{q}_y$ and $\dot{\mathbf{d}} = \mathbf{q}_d$

The coordinates evolve as (22) and (23) until any of y_{nt}, d_{kt} reaches zero, at which point the velocity changes discontinuously. Note that all the coordinates d_{kt} can reach zero, but only those y_{nt} that have $u_{y,nt} > |\mathbf{z}_n \cdot \mathbf{s}_t|$ can do so. Let us consider each case:

- $y_{nt} = 0$

  Since the particle is constrained by $y_{nt} \geq 0$, the velocity is reflected as

  $$\dot{y}_{nt} \to -\dot{y}_{nt} \qquad (24)$$

- $d_{kt} = 0$

  Let us call τ_{kt}^- and τ_{kt}^+ the times just before and after the wall hit. If $d_{kt}(\tau < \tau_{kt}) < 0$, imposing conservation of energy at both sides of the $d_{kt} = 0$ wall gives,

  $$\frac{\dot{d}_{kt}^2(\tau_{kt}^+)}{2} = \frac{\dot{d}_{kt}^2(\tau_{kt}^-)}{2} + \Delta_1 \qquad (25)$$

  where

  $$\Delta_1 = U_1(s_{kt} = 0) - U_1(s_{kt} = 1).$$

  If $d_{kt}(\tau < \tau_{kt}) > 0$ we invert the roles of τ_{kt}^- and τ_{kt}^+ in the above equation. If $\dot{d}_{kt}^2(\tau_{kt}^+) > 0$, the particle crosses the wall with this new velocity for $\dot{d}_{kt}$, and if $\dot{d}_{kt}^2(\tau_{kt}^+) < 0$, it gets reflected with $\dot{d}_{kt}(\tau_{kt}^+) = -\dot{d}_{kt}(\tau_{kt}^-)$.

B.2 Models 2 and 3

In these cases we are interested in sampling from (11),

$$\begin{aligned} &p(\mathbf{y}_t, \mathbf{v}_t, \mathbf{s}_t | \mathbf{x}_t, \mathbf{Z}, \mathbf{a}) \qquad (26) \\ &\propto p(\mathbf{x}_t, \mathbf{y}_t | \mathbf{v}_t, \mathbf{z}) p(\mathbf{v}_t) p(\mathbf{s}_t | a) \\ &\propto \prod_{n \in \mathcal{O}_t} e^{-\frac{1}{2}(y_{nt} - x_{nt}\mathbf{z}_n \cdot \mathbf{f}_t)^2} e^{-\frac{|\mathbf{v}_t|^2}{2}} \prod_k a_k^{s_{kt}}(1 - a_k)^{1 - s_{kt}} \end{aligned}$$

with $y_{nt} \geq 0$, and for Model 3 also have the constraint $v_{kt} \geq 0$. As in Model 1, we augment the binary variables $\mathbf{s}_t$ with continuous variables $\mathbf{d}_t$. The resulting piecewise continuous distribution is

$$\begin{aligned} U_2 &= -\log p(\mathbf{y}_t, \mathbf{v}_t, \mathbf{d}_t | \theta, \mathbf{x}_t) \qquad (27) \\ &= \frac{1}{2}\mathbf{w}_t^T D_t \mathbf{w}_t - \mathbf{r} \cdot \mathbf{w}_t + \frac{\mathbf{d}_t \cdot \mathbf{d}_t}{2} + f(\mathbf{s}_t) + const \end{aligned}$$

where $f(\mathbf{s}_t)$ was defined in (19) and

$$\mathbf{w}_t = \begin{pmatrix} \mathbf{v}_t \\ \mathbf{y}_t \end{pmatrix} \in \mathbb{R}^{K+N_t}, \qquad (28)$$

with N_t the number of elements in $\mathcal{O}_t$. The coefficient of the linear term for $\mathbf{w}_t$ in (27) is

$$\mathbf{r} = \begin{pmatrix} \mathbf{r}_v \\ \mathbf{r}_y \end{pmatrix} \in \mathbb{R}^{K+N_t} \qquad (29)$$

where

$$r_{v,k} = -s_{kt}\sum_n z_{n(K+1)}z_{nk} \qquad k=1,\ldots,K \quad (30)$$

$$r_{y,n} = z_{n(K+1)}x_{nt} \qquad n\in\mathcal{O}_t \quad (31)$$

and the coefficient of the quadratic term in (27) is

$$D_t = M_t^T M_t \quad \in \quad \mathbb{R}^{(K+N_t)\times(K+N_t)} \quad (32)$$

where

$$M_t = \begin{pmatrix} J_t & \mathbb{I}_{N_t} \\ \mathbb{I}_K & 0 \end{pmatrix}$$

with $J_t \in \mathbb{R}^{N_t\times K}$,

$$(J_t)_{n,k} = -x_{nt}s_{kt}z_{nk}\,.$$

A sample $(\mathbf{y}_t, \mathbf{v}_t, \mathbf{d}_t)$ from (27) gives a sample $(\mathbf{y}_t, \mathbf{v}_t, \mathbf{s}_t)$ from the original distribution (26) using the rule (20). In order to sample from (27) using HMC, we introduce momentum variables $\mathbf{q}_t$ and $\mathbf{q}_d$ and consider the Hamiltonian

$$H_2 = U_2 + K_w + \frac{\mathbf{q}_d\cdot\mathbf{q}_d}{2} \quad (33)$$

where

$$K_w = \frac{1}{2}\mathbf{q}_w^T D_t^{-1}\mathbf{q}_w. \quad (34)$$

One can verify that $\det(D_t) = 1$, so there is no term proportional to $\log(\det(D_r))$ in K_w. Since $\mathbf{q}_d = \dot{\mathbf{d}}$ and $\mathbf{q}_{\mathbf{w}_t} = D_t\dot{\mathbf{w}}_t$, as follows from (21), in each iteration we sample the initial velocities from[5]

$$\begin{aligned} \dot{d}_{kt}(0) &\sim \mathcal{N}(0,1)\,, \\ \dot{\mathbf{w}}_t(0) &\sim \mathcal{N}(0, D_t^{-1})\,. \end{aligned} \quad (35)$$

The solution to the equations of motion for w_{ti} is

$$w_{ti}(\tau) = \mu_{ti} + \rho_{ti}\cos(\tau) + \dot{w}_{ti}(0)\sin(\tau) \quad (36)$$

$$= \mu_{ti} + u_{ti}\sin(\tau+\phi_{ti}) \quad (37)$$

with

$$\boldsymbol{\mu}_t = D_t^{-1}\mathbf{r}\,, \quad (38)$$

$$\rho_{ti} = w_{ti}(0) - \mu_{ti} \quad (39)$$

$$u_{ti} = \sqrt{\rho_{ti}^2 + \dot{w}_{ti}(0)} \quad (40)$$

$$\phi_{ti} = \tan^{-1}(\rho_{ti}/\dot{w}_{ti}(0)) \quad (41)$$

[5]In general, HMC algorithms sample initial *momenta* [Neal, 2010], but since we have exact solutions of the equations of motion in terms of initial velocities, sampling the latter is more efficient. This approach was used before in [Pakman and Paninski, 2013a, Lan et al., 2012].

while the solutions for d_{kt} are the same as in Model 1, see (23). To sample from (35) and to compute (38), note that

$$D_t^{-1} = Z_t^T Z_t \quad (42)$$

with

$$Z_t = \begin{pmatrix} \mathbb{I}_K & -J_t^T \\ 0 & \mathbb{I}_{N_t} \end{pmatrix}, \quad (43)$$

so we can sample $\boldsymbol{\varepsilon}\sim\mathcal{N}(0,1)$ and define

$$\dot{\mathbf{w}}_t(0) = Z_t\boldsymbol{\varepsilon}. \quad (44)$$

The coordinates evolve as (23) and (37) until any of w_{ti} or d_{kt} reaches zero, at which point the velocity changes discontinuously. Similarly to Model 1, all of the coordinates d_{kt} can reach zero, but only those w_{ti} with $u_{ti}\geq|\mu_{ti}|$ can do so. Also, in Model 2 we only consider $w_{ti}=0$ for $i>K$. Let us consider each case:

- $w_{ti}=0$

  In this case the velocity is reflected off the $w_{ti}=0$ wall. Let us define the vector $\mathbf{h}_i\in\mathbb{R}^{K+N_t}$ as

  $$h_{i,j} = \delta_{i,j} \qquad j=1,\ldots,K+N_t.$$

  As we show in section B.3, the reflected velocity is given by

  $$\dot{\mathbf{w}}_t^R = \dot{\mathbf{w}}_t - 2\alpha D_t^{-1}\mathbf{h}_i \quad (45)$$

  where

  $$\alpha = \frac{\mathbf{h}_i\cdot\dot{\mathbf{w}}_t}{\mathbf{h}_i^T D_t^{-1}\mathbf{h}_i}\,. \quad (46)$$

- $d_{kt}=0$

  Here the situation is similar to Model 1, with Δ_1 in (25) replaced by

  $$\Delta_2 = U_2(s_{kt}=0) - U_2(s_{kt}=1) + K_w(s_{kt}=0) - K_w(s_{kt}=1).$$

In both cases, after the velocity changes, we update the values of (38)-(41) and then continue the trajectory.

B.3 Velocity Reflection in Models 2 and 3

We now derive equations (45)-(46) for the reflected value of $\dot{\mathbf{w}}_t$ in Models 2 and 3. To simplify the notation we omit the subindex t. Consider the $\mathbf{w}$-dependent terms in the Hamiltonian,

$$H_w = \frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^T D(\mathbf{w}-\boldsymbol{\mu}) + \frac{1}{2}\dot{\mathbf{w}}^T D\dot{\mathbf{w}}$$

and the constraint

$$\mathbf{w} \cdot \mathbf{h}_i \geq 0. \tag{47}$$

Remember from (32) that $D = M^T M$. Making a change of coordinates

$$\mathbf{g} = M\mathbf{w} - M\boldsymbol{\mu}$$

we get

$$H_w = \frac{1}{2}\mathbf{g} \cdot \mathbf{g} + \frac{1}{2}\dot{\mathbf{g}} \cdot \dot{\mathbf{g}}$$

and the constraint (47) becomes

$$\mathbf{g} \cdot \tilde{\mathbf{h}}_i + \boldsymbol{\mu} \cdot \mathbf{h}_i \geq 0\,,$$

where $\tilde{\mathbf{h}}_i = (M^{-1})^T \mathbf{h}_i$. In this frame, when the equality is satisfied and the particle bounces off the wall, the reflected velocity is [Pakman and Paninski, 2013a]

$$\dot{\mathbf{g}}_R = \dot{\mathbf{g}} - 2\alpha\tilde{\mathbf{h}}_i \tag{48}$$

with

$$\begin{aligned} \alpha &= \frac{\dot{\mathbf{g}} \cdot \tilde{\mathbf{h}}_i}{\|\tilde{\mathbf{h}}_i\|^2} \\ &= \frac{\mathbf{h}_i \cdot \dot{\mathbf{w}}_t}{\mathbf{h}_i^T D^{-1} \mathbf{h}_i}. \end{aligned}$$

Multiplying (48) on the left by M^{-1}, we get the reflected velocity in the original frame

$$\dot{\mathbf{w}}^R = \dot{\mathbf{w}} - 2\alpha D^{-1}\mathbf{h}_i\,.$$

It is easy to check that

$$\dot{\mathbf{w}}^T D \dot{\mathbf{w}} = \dot{\mathbf{w}}^{R,T} D \dot{\mathbf{w}}^R\,,$$

so the energy is conserved in the reflection.

Acknowledgements

This work was supported by an NSF CAREER award and by the US Army Research Laboratory and the US Army Research Office under contract number W911NF-12-1-0594.

References

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

E.I. George and R.E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

M.D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Arxiv preprint arXiv:1111.4246*, 2011.

Charles Kemp and Joshua B Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008.

Lukasz A Kurgan, Krzysztof J Cios, Ryszard Tadeusiewicz, Marek R Ogiela, and Lucy S Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine*, 23:149, 2001.

Shiwei Lan, Vassilios Stathopoulos, Babak Shahbaba, and Mark Girolami. Lagrangian Dynamical Monte Carlo. *arXiv preprint arXiv:1211.3759*, 2012.

Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

Shakir Mohamed. Generalised Bayesian Matrix Factorisation Models. *PhD Thesis, Cambdridge University*, 2012.

Shakir Mohamed, Katherine A Heller, and Zoubin Ghahramani. Bayesian Exponential Family PCA. In *NIPS 2008*, pages 1089–1096, 2008.

Shakir Mohamed, Katherine Heller, and Zoubin Ghahramani. Bayesian and L1 Approaches for Sparse Unsupervised Learning. ICML '12, pages 751–758, 2012.

Radford M Neal. Learning stochastic feedforward networks. *Department of Computer Science, University of Toronto*, 1990.

Radford M Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

R.M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Ari Pakman and Liam Paninski. Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians. *Journal of Computational and Graphical Statistics, arXiv:1208.4118*, 2013a.

Ari Pakman and Liam Paninski. Auxiliary-variable Exact Hamiltonian Monte Carlo Samplers for Binary Distributions. *arXiv:1311.2166, NIPS*, 2013b.

Jonathan W Pillow and James G Scott. Fully Bayesian inference for neural models with negative-binomial spiking. In *NIPS*, pages 1907–1915, 2012.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using Markov

chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

Michael A Shwe, B Middleton, DE Heckerman, M Henrion, EJ Horvitz, HP Lehmann, and GF Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Meth. Inform. Med*, 30:241–255, 1991.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. 1986.

Ziyu Wang, Shakir Mohamed, and Nando de Freitas. Adaptive Hamiltonian and Riemann Manifold Monte Carlo Samplers. ICML '13, 2013.

Mingyuan Zhou, Lauren Hannah, David B Dunson, and Lawrence Carin. Beta-Negative Binomial Process and Poisson Factor Analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 1462–1471, 2012.