Markov Latent Feature Models

Aonan Zhang AZ2385@COLUMBIA.EDU
John Paisley JPAISLEY@COLUMBIA.EDU
Department of Electrical Engineering & Data Science Institute
Columbia University, New York, NY, USA

Abstract

We introduce Markov latent feature models (MLFM), a sparse latent feature model that arises naturally from a simple sequential construction. The key idea is to interpret each state of a sequential process as corresponding to a latent feature, and the set of states visited between two null-state visits as picking out features for an observation. We show that, given some natural constraints, we can represent this stochastic process as a mixture of recurrent Markov chains. In this way we can perform *correlated latent feature modeling* for the sparse coding problem. We demonstrate two cases in which we define finite and infinite latent feature models constructed from first-order Markov chains, and derive their associated scalable inference algorithms. We show empirical results on a genome analysis task and an image denoising task.

1. Introduction

Latent feature models learn the unobserved factors that are shared among a collection of objects. Often a small fraction of these latent features can be used to jointly describe, or "sparsely code," a single object. This assumption is often made for a wide range of tasks (Ghahramani et al., 2007). For example, each DNA sequence can be assigned features that measure repeating patterns of certain base pairs, or each image can be assigned features that correspond to the items it contains.

The process for making latent feature assignments can be thought of as generating a 0-1 matrix where the 1-elements in each row index the latent features assigned to the object. For example, the Indian buffet process (IBP) (Griffiths & Ghahramani, 2011) defines a feature allocation whereby features are independently assigned according to a rich-get-richer scheme. The IBP is a Bayesian nonparametric model in which the number of latent features can grow to infinity with data. It also assumes exchangeability among objects, meaning the order of the data does not affect the feature assignment probabilities. With such assumptions there always exists a mixing measure (de Finetti's measure) by which feature allocation scheme for different objects are conditionally independent. For example, the mixing measure for the IBP is the beta process (Thibaux & Jordan, 2007). Employing such representations allows for simple variational inference algorithms that can easily scale to large data sets (Hoffman et al., 2013; Sertoglu & Paisley, 2015; Shah et al., 2015).

Models based on the IBP and beta process assume independence among the latent features allocated to an object, but in many cases these latent features may have dependencies. For example, in a natural image a car is more likely to co-occur with a bus rather than a whale. Several methods have been developed for modeling dependencies among latent features or clusters. For example, beta diffusion trees (Heaukulani et al., 2014) organize latent features in a tree to learn a multi-resolution feature structure. Other tree models, such as the nested Chinese restaurant processes (Blei et al., 2010) and the nested hierarchical Dirichlet processes (Paisley et al., 2015), use a discrete-path based tree structure to select latent features for each object. To avoid the rigid structure of trees in a mixed-membership framework, Markov mixed-membership models (Zhang & Paisley, 2015) propose instead modeling the pair-wise correlation of latent clusters with a Markov random walk on a fully-connected, finite graph.

We propose a *Markov latent feature model* (MLFM), which extends the idea of using Markov random walks to latent factor modeling problems such as those addressed by the IBP and beta process. The main novelty in this new framework is that we use a sequential block—a subsequence between two adjacent visits to a "null state"—to define the feature allocation process (Section 2). Since only a subset of states will be visited prior to returning to the null state,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

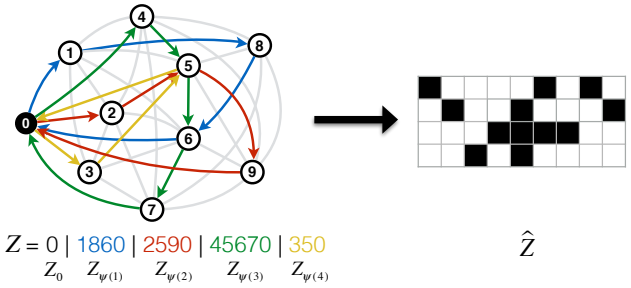

Figure 1. An illustration of the construction of a 0-1 matrix from a sequential process, which we define to be a mixture of recurrent Markov chains. On the LHS, the chain $\mathbf{Z}$ starts from a null state $Z_0 = 0$ and generates four blocks (subsequences) $\mathbf{Z}_{\psi(1)}$ to $\mathbf{Z}_{\psi(4)}$ by returning to 0 four times (shown as four colored paths on the graph). One the RHS, this sequence constructs a 0-1 matrix with four rows and the columns indicating the unique set of states visited in each block.

a MLFM is a sparse coding model. We introduce two scalable MLFM models, a parametric and nonparametric version, for which we directly define the mixing measure of the associated recurrent Markov chain (Section 3). This allows us to derive a scalable variational inference algorithm (Section 4). Finally, we apply MLFM to a genome analysis task and an image denoising task to show its effectiveness (Section 5).

2. Feature Allocation via Sequences

Before describing the specific generative processes we use, we discuss the central property of the proposed Markov latent feature model (MLFM) that distinguishes it from other approaches to sparse coding. Let $\mathbf{Z} = (Z_0, Z_1, \cdots)$ be an infinitely long stochastic process, where each $Z_i \in \mathbb{N} \cup \{0\}$ and $Z_0 = 0$, with 0 indexing the "null state". As we shall see, the null state plays the role of partitioning latent features for different objects, while $\mathbb{N}$ is a feature index set.

The generative process we define sequentially pick features for an object until the process returns to 0. Let $\tau(0) = 0$ be the index of the initial null state and

$$\tau(1) = \min\{n > \tau(0) \ : \ Z_n = 0\}. \tag{1}$$

The sequence through $Z_{\tau(1)}$ selects the features assigned to the first object as the unique set of states visited between $Z_{\tau(0)}$ and $Z_{\tau(1)}$. We call $\tau(1)$ the first return time and define $\psi(1)$ to be the sequence $(\tau(0)+1, \ldots, \tau(1))$. Therefore $\mathbf{Z}_{\psi(1)}$ is the first *block* of the process $\mathbf{Z}$ and corresponds to the features allocated to the first observation.

We continue this procedure through a second return time $\tau(2)$, constructing the second subsequence $\mathbf{Z}_{\psi(2)}$ from which we obtain the set of latent features assigned to the second observation. More generally, if we have N observations, then we read the stochastic process until step $\tau(N)$, the time we finish selecting features for the last object.

We can use this set of blocks to construct a 0-1 matrix $\widehat{\mathbf{Z}}$, where each row indicates the features associated with the corresponding observation similar to the IBP. We show an example for $N = 4$ in Figure 1.

Thus far we have not defined the distribution of $\mathbf{Z}$. However, we choose to make the following two restrictions:

1. The null state should be visited infinitely many times.
2. The rows of $\widehat{\mathbf{Z}}$ should be exchangeable.

The first restriction allows us to model an infinite number of observations and is a statement about the *recurrency* of $\mathbf{Z}$. The second restriction is made to allow for simple inference using a mixing measure. Our added goal of modeling feature correlations leads us to enforce the second restriction via *Markov exchangeability*. Recall that a sequence $\mathbf{Z}$ is Markov exchangeable if the probability of two sequences $\mathbf{Z}'$ and $\mathbf{Z}''$ is the same when they share a permuted collection of subsequences. In the context of Figure 1, this simply states that the probability of $\mathbf{Z}$ is the same according to the chosen distribution if we permute a finite number of $Z_{\psi(i)}$. The following lemma is a direct result of these definitions.

Lemma 1. *The rows of $\widehat{\mathbf{Z}}$ are exchangeable if $\mathbf{Z}$ is Markov exchangeable.*

Theorem 1. *(Diaconis & Freedman, 1980) A recurrent process $\mathbf{Z}$ is Markov exchangeable if and only if it is a mixture of Markov chains.*

We therefore specify $\mathbf{Z}$ as a mixture of *recurrent Markov chains*. As a result, the latent features are correlated, which can be viewed as a graph where the edges indicate Markov transitions among states (see Figure 1). We observe that models such as the IBP satisfy the above requirements, but without modeling correlations. Using the Markov property provides this modeling capacity in a way that allows for simple inference and has straightforward nonparametric extensions.

3. Markov Latent Feature Models

In Section 2 we proposed restricting $\mathbf{Z}$ to be a mixture of recurrent Markov chains. In principle, any formulation based on this restriction would be valid, including those whose mixing measure is unknown, but for practical purposes we would like to explicitly model the mixing measure of the recurrent Markov chain. We therefore propose two models below, one parametric and one nonparametric, both based on a simple first-order Markov assumption.

3.1. A parametric model

Assume we have N observations and have $K+1$ possible states (including the null state). We can formulate $\mathbf{Z}$ as

$$p(\mathbf{Z}_{1:\tau(N)}|Z_0) = \int \prod_{j=0}^{\tau(N)-1} p(Z_{j+1}|Z_j,\boldsymbol{\theta})\mu(d\boldsymbol{\theta}), \quad (2)$$

where $p(Z_{j+1}|Z_j,\boldsymbol{\theta}) = \theta_{Z_j,Z_{j+1}}$. We let the mixing measure μ be a prior on the Markov transition matrix $\boldsymbol{\theta}$. In particular, we let the vector $\boldsymbol{\theta}_k = (\boldsymbol{\theta}_{k,0},\ldots,\boldsymbol{\theta}_{k,K})$ be distributed as

$$\boldsymbol{\theta}_k \sim \text{Dir}\big(\tfrac{\alpha}{K+1},\cdots,\tfrac{\alpha}{K+1}\big), \quad 0 \le k \le K. \quad (3)$$

Notice that together with the null state we have $K+1$ states; we do not separate the null state from the other states, but jointly model them together.

We call this model the finite Markov latent feature model (MLFM in our experiments). When the expected return time to the null state is much smaller than K, MLFM will be a sparse coding model in that each observation will possess a small subset of features. Unlike models based on the beta process, in which the number of features for each observation follows a Poisson distribution, the distribution on the number of features in MLFM cannot be derived in closed-form. However, this distribution is connected to the stationary distribution of the process. We next present a bound on the expected number of features.

Proposition 1. *Suppose $\boldsymbol{\theta}$ is known and Δ is its stationary distribution. Let $\mathcal{A}_i$ be the set of unique features used by object i, then $\mathbb{E}[|\mathcal{A}_i|] \le \frac{1}{\Delta_0}$, where Δ_0 corresponds to the null state.*

Proof. First observe that $|\mathcal{A}_i| \le \tau(i)-\tau(i-1)$ because the sequence may return to the same feature multiple times. By the Markov exchangeablity of $\mathbf{Z}$, we have $|\mathcal{A}_i| =_d |\mathcal{A}_1|$ and $\tau(i)-\tau(i-1) =_d \tau(1)$. Since $\theta_{k,k'} > 0$, the Markov chain is regular and thus ergodic. Since $\mathbb{E}[\tau(1)]$ is the expected return time for a regular Markov chain and $\mathbb{E}[\tau(1)] = \frac{1}{\Delta_0}$ (Norris, 1998), the result follows. □

3.2. A nonparametric model

In our second example, we extend the above model to an infinite number of features in the spirit of the IBP and beta process. We use the hierarchical Dirichlet processes (HDP) (Teh et al., 2006) to model the mixing measure $\boldsymbol{\theta}$:

$$\boldsymbol{\beta} \sim \text{GEM}(\alpha), \quad \boldsymbol{\theta}_k \sim \text{Dir}(\alpha\boldsymbol{\beta}), \quad (4)$$

where $\beta_k = v_k \prod_{k'=0}^{k-1}(1-v_{k'})$ and $v_{k'} \sim \text{Beta}(1,\alpha)$. We call this model the infinite Markov latent feature model (iMLFM in the experiments).

3.3. Application to a linear Gaussian model

We apply these models to the dictionary learning problem using a linear Gaussian model. In this case, the data matrix of N observations $\mathbf{X} = [\mathbf{X}_1,\cdots,\mathbf{X}_N]$ is modeled as

$$\mathbf{X} = \mathbf{W}(\widehat{\mathbf{Z}}^\top \circ \mathbf{C}) + \boldsymbol{\epsilon}, \quad (5)$$

where $\mathbf{W} = [\mathbf{w}_1,\cdots,\mathbf{w}_K]$ is the dictionary matrix with K elements, $\mathbf{C}$ is a $K \times N$ matrix, $\boldsymbol{\epsilon} = [\epsilon_1,\cdots,\epsilon_N]$ is a noise matrix, and

$$w_k \sim \mathcal{N}(\mathbf{0},\tfrac{1}{\eta}I), \quad c_{ki} \sim \mathcal{N}(0,\tfrac{1}{\lambda}), \quad \epsilon_i \sim \mathcal{N}(\mathbf{0},\sigma^2 I).$$

As defined above, the coding matrix $\widehat{\mathbf{Z}}$ is generated from the Markov sequence $\mathbf{Z}$ described in Section 2 using the priors on either the finite or infinite state Markov chain define above. In the infinite-state model, $K = \infty$.

3.4. Discussion

The two examples of an MLFM given above are models that induce *directed* correlations among latent features. However, they are not the only choice. From Theorem 1, we can apply any mixture of recursive Markov chains to build a model. Another choice would be to use an edge-reinforced random walk (ERRW). An ERRW-induced latent feature model is *undirected*. A class of ERRWs has been shown to be a mixture of reversible Markov chains, for which Bayesian inference has recently been studied (Diaconis & Rolles, 2006; Bacallado et al., 2013). However, constructing nonparametric priors for reversible Markov chains is non-trivial. Knowles et al. (2014) provides one solution, but a scalable version has not yet been developed.

Several previous works have studied the theoretical properties of other feature allocation constructions. For example, Broderick et al. (2013) studied an exchangeable class of feature partitions using a "paintbox" characterization, while Heaukulani & Roy (2013) analyzed the combinatorial structure of beta negative binomial processes and Zhou et al. (2015) investigate such constructions for feature count matrices.

4. Inference

We derive a variational inference algorithm for the parametric Markov latent feature model where we model the mixing measure as a Dirichlet distribution. We can extend inference to the nonparametric case by modeling $\boldsymbol{\beta}$ in Equation (4) using, e.g., the direct assignment method as mentioned in Liang et al. (2007); Johnson & Wilsky (2014) for the HDP; another recent fully Bayesian method is Zhang et al. (2016). We exclude this additional step in our algorithm below since the derivation is identical to the HDP in this portion of the model.

Algorithm 1 Sparse coding with greedy search

Input: $q(\boldsymbol{\theta})$ and $\mathbf{W}$.
for $i = 1$ **to** N **do**
  1. Set $\mathbf{Z}_{\psi(i)} = \emptyset$ and $\mathcal{A}_i = \emptyset$.
  while $\max_j \xi_j > 0$ **do**
    (a) Set $\xi_j = \mathcal{L}_{\mathcal{A}_i}([\mathbf{Z}_{\psi(i)}, j, 0]) - \mathcal{L}_{\mathcal{A}_i}([\mathbf{Z}_{\psi(i)}, 0])$.
    (b) Set $j' = \arg\max_j \xi_j$.
    (c) Set $\mathbf{Z}_{\psi(i)} \leftarrow [\mathbf{Z}_{\psi(i)}, j']$ and $\mathcal{A}_i \leftarrow \mathcal{A}_i \cup \{j'\}$.
  end while
  Set $\mathbf{Z}_{\psi(i)} = [\mathbf{Z}_{\psi(i)}, 0]$.
end for

4.1. Batch Variational Inference

The joint distribution of the Markov latent feature model factorizes as

$$p(\boldsymbol{\theta}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{X}) = \left[\prod_{j=0}^{K} p(\boldsymbol{\theta}_j) p(\mathbf{w}_j)\right] \times \left[\prod_{i=1}^{N} p(\mathbf{C}_i) p(\mathbf{Z}_{\psi(i)}|\boldsymbol{\theta}) p(\mathbf{X}_i|\mathbf{C}_i, \mathbf{Z}_{\psi(i)}, \mathbf{W})\right].$$

We recall that $\mathbf{Z}_{\psi(i)}$ is the block of the Markov chain that selects features for the ith observation and must terminate at the null state. We restrict the posterior to a factorized form as well,

$$q(\boldsymbol{\theta}, \mathbf{W}, \mathbf{C}, \mathbf{Z}) = \left[\prod_{j=1}^{K} q(\boldsymbol{\theta}_j) q(\mathbf{w}_j)\right] \left[\prod_{i=1}^{N} q(\mathbf{C}_i) q(\mathbf{Z}_{\psi(i)})\right],$$

and we define

$$q(\boldsymbol{\theta}_j) = \mathrm{Dir}(\mathbf{a}_j), \qquad q(\mathbf{w}_j) = \delta_{\mathbf{w}_j}(\cdot),$$
$$q(\mathbf{C}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad q(\mathbf{Z}_{\psi(i)}) = \delta_{\mathbf{Z}_{\psi(i)}}(\cdot). \tag{6}$$

The variational objective is

$$\mathcal{L} = \mathbb{E}_q[\ln p(\boldsymbol{\theta}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{X})] - \mathbb{E}_q[\ln q(\boldsymbol{\theta}, \mathbf{W}, \mathbf{C}, \mathbf{Z})].$$

Using this factorization, we observe in advance that our algorithm below is equivalent to a MAP-EM algorithm for maximizing $p(\mathbf{X}, \mathbf{W}, \mathbf{Z})$, where $\mathbf{C}$ and $\boldsymbol{\theta}$ constitute the hidden data. This is because

$$q(\boldsymbol{\theta}, \mathbf{W}, \mathbf{C}, \mathbf{Z}) = q(\boldsymbol{\theta}, \mathbf{C}|\mathbf{W}, \mathbf{Z}) q(\mathbf{W}, \mathbf{Z})$$

and $\boldsymbol{\theta}$ and $\mathbf{C}$ are conditionally independent and can be solved exactly given the point estimates $\mathbf{W}$ and $\mathbf{Z}$, which the delta q distribution enforces. In other words, the mean-field representation for $\boldsymbol{\theta}$ and $\mathbf{C}$ is exact and not an approximation in this case.

Update Z and C: Sparse coding with greedy search. We jointly update $\mathbf{Z}_{\psi(i)}$ and $\mathbf{C}_i$ using a new approach to sparse coding with Bayesian models. The method is similar to orthogonal matching pursuits, used in sparse coding by K-SVD (Aharon et al., 2006), in that it greedily selects the next feature to add by integrating out the corresponding weight, followed by an update of all weights on the active features. The structure of the algorithm is also similar to MAP-EM inference for mixtures of factor analyzers (Ghahramani & Hinton, 1996).

To sparsely code the ith observation, we can focus on the objective term

$$\mathcal{L}(\mathbf{Z}_{\psi(i)}, q(\mathbf{C}_i)) = \mathbb{E}_q\left[\ln \frac{p(\mathbf{X}_i, \mathbf{C}_i, \mathbf{Z}_{\psi(i)}|\boldsymbol{\theta}, \mathbf{W})}{q(\mathbf{C}_i)}\right]. \tag{7}$$

First observe that given $\mathbf{Z}_{\psi(i)}$,

$$q(\mathbf{C}_i) = p(\mathbf{C}_i|\mathbf{X}_i, \mathbf{Z}_{\psi(i)}, \mathbf{W}) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{8}$$

is known exactly and is a multivariate Gaussian derived explicitly below. Here, we jointly update $\mathbf{Z}_{\psi(i)}$ and $q(\mathbf{C}_i)$ by incrementally extending (or terminating) the path $\mathbf{Z}_{\psi(i)}$.

Because our inference problem is equivalent to MAP-EM for the joint likelihood $p(\mathbf{X}, \mathbf{W}, \mathbf{Z})$ we can do this as follows: Let $\mathcal{A}_i$ be the current set of features selected by the path $\mathbf{Z}_{\psi(i)}$ and $q(\mathbf{C}_{\mathcal{A}_i}) = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{A}_i}, \boldsymbol{\Sigma}_{\mathcal{A}_i})$ be the corresponding marginal posterior over these dimensions, where

$$\boldsymbol{\Sigma}_{\mathcal{A}_i} = (\lambda I + \tfrac{1}{\sigma^2} \mathbf{W}_{\mathcal{A}_i}^\top \mathbf{W}_{\mathcal{A}_i})^{-1}, \; \boldsymbol{\mu}_{\mathcal{A}_i} = \tfrac{1}{\sigma^2} \boldsymbol{\Sigma}_{\mathcal{A}_i} \mathbf{W}_{\mathcal{A}_i}^\top \mathbf{X}_i.$$

We expand the path of $\mathbf{Z}_{\psi(i)}$ using EM by constructing

$$\mathcal{L}_{\mathcal{A}_i}(\mathbf{Z}_{\psi(i)}) = \mathbb{E}_q[\ln p(\mathbf{X}_i, \mathbf{Z}_{\psi(i)}|\mathbf{C}_{\mathcal{A}_i}, \mathbf{W}, \boldsymbol{\theta})]$$

where the expectation is over $\boldsymbol{\theta}$ and the subset of $\mathbf{C}_i$ indexed by $\mathcal{A}_i$ using $q(\mathbf{C}_{\mathcal{A}_i})$ derived above. The remaining dimensions of $\mathbf{C}_i$ are marginalized out *a priori*. Since we are dealing with multivariate normal variables, all calculations are in closed form and remain Gaussian.

We then greedily pick the next state that improves this objective the most and add that state to the end of the path $\mathbf{Z}_{\psi(i)}$, or we terminate if moving to the null state and adding no more features provides the best improvement. When the algorithm terminates, we have a point estimate of the path $\mathbf{Z}_{\psi(i)}$ that selects the latent features for the ith observation, and the corresponding conditional posterior q distribution on the weight vector $\mathbf{C}_i$. By the equivalent MAP-EM construction of this algorithm, each step is guaranteed to increase the objective function. We summarize this greedy algorithm in Algorithm 1.

Proposition 2. *The sparse coding greedy search algorithm will stop in a finite number of steps.*

See the appendix for the proof. Furthermore, we observe in our experiments that the algorithm tends to terminate after a small fraction of available features have been selected, using a number comparable to IBP-based models.

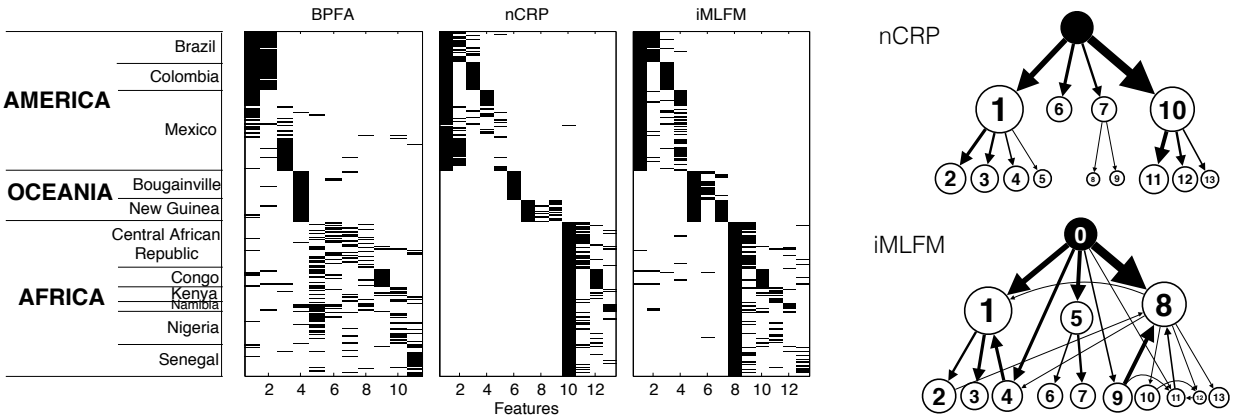

Figure 2. (Left) Factors learned from BPFA, nCRP, and iMLFM on the HGDP-CEPH dataset. Observations from various regions are aligned vertically, as displayed on the left side. (Right) Graph learned from iMLFM and nCRP.

Update $q(\boldsymbol{\theta})$: The distribution $q(\boldsymbol{\theta}_j) = \mathrm{Dir}(\mathbf{a}_j)$ can be found by optimizing the Markov chain portion of the objective function below,

$$\mathcal{L}(q(\boldsymbol{\theta})) = \mathbb{E}_q[\ln p(\boldsymbol{\theta})] + \textstyle\sum_{i=1}^{N} \mathbb{E}_q[\ln p(\mathbf{Z}_{\psi(i)}|\boldsymbol{\theta})].$$

Since $\mathbf{Z}_{\psi(i)}$ is a point estimate, this is equivalent to finding the conditional posterior of $\boldsymbol{\theta}_j$, and thus

$$\mathbf{a}_{j,j'} = \frac{\alpha}{K+1} + \sum_{i=0}^{\tau(N)-1} \mathbb{1}(Z_i = j, Z_{i+1} = j').$$

Update $\mathbf{W}$: For this point estimate, we want to maximize

$$\mathcal{L}(\mathbf{W}) = \ln p(\mathbf{W}) + \textstyle\sum_{i=1}^{N} \mathbb{E}_q[\ln p(\mathbf{X}_i|\mathbf{C}_i, \widehat{\mathbf{Z}}_i, \mathbf{W})].$$

Let $q(\mathbf{C}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, and define $\widetilde{\mathbf{Z}}_i = \mathrm{diag}(\widehat{\mathbf{Z}}_i)$. We can differentiate with respect to $\mathbf{W}$ to find that

$$\mathbf{W} = \Big[\sum_{i=1}^{N} \mathbf{X}_i \boldsymbol{\mu}_i^\top \widetilde{\mathbf{Z}}_i\Big] \Big[\eta\sigma^2 I + \sum_{i=1}^{N} \widetilde{\mathbf{Z}}_i(\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top + \boldsymbol{\Sigma}_i)\widetilde{\mathbf{Z}}_i\Big]^{-1}.$$

4.2. Stochastic Variational Inference

We also derive a stochastic inference algorithm for large scale learning. We give the corresponding modifications for this algorithm in the appendix. The result is effectively a stochastic EM algorithm that shares one update with SVI (Hoffman et al., 2013) for learning $q(\boldsymbol{\theta})$ and uses stochastic gradient directly on the dictionary $\mathbf{W}$.

5. Experiments

We demonstrate the effectiveness of our MLFM framework on two tasks. The first task is an analysis of the HGDP-CEPH cell line panel dataset (Rosenberg et al., 2002) for which we test batch learning performance of our two models. The second task is image denoising, where batch learning is slower and so we use stochastic inference.

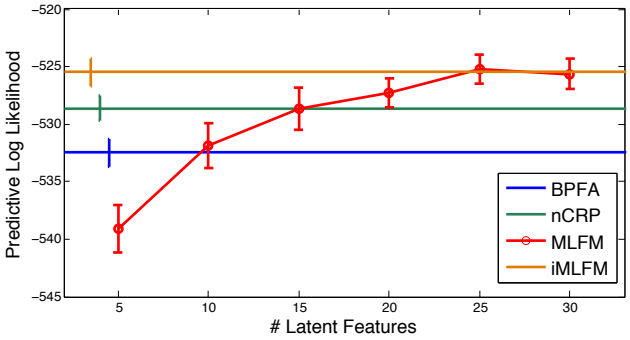

Figure 3. Average predictive result on HGDP-CEPH dataset.

5.1. HGDP-CEPH Cell Line Panel

For this small-scale experiment, we use a subset of 266 individuals across 11 countries from the HGDP-CEPH Human Genome Diversity Cell Line Panel (Rosenberg et al., 2002)[1]. Each person is represented by their genotypes measured at $D = 377$ autosomal microsatellite loci.

We split this data into a set of 54 individuals for testing, and use the rest for training. For evaluation we use average predictive log-likelihood of the testing set, which we approximated using Monte Carlo integration over the q distributions. We experiment with MLFM letting K range from 5 to 30 features, and iMLFM truncated to 100 latent features. We compare with BPFA (Paisley & Carin, 2009) and nested CRP (Blei et al., 2010), which we modified into a linear Gaussian model. Both models were truncated to 100 latent features as well. We set hyper-parameters to be $\eta = 1, \lambda = 1, \sigma = 0.8$. We ran each model 20 times and averaged the results. All models converge by 100 global

[1] As reported in (Rosenberg et al., 2002), the remaining individuals form two large heterogeneous clusters which are hard to distinguish in general.

Table 1. PSNR | SSIM (average # features used per patch) on various images and noise settings.

BARBARA	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$
iMLFM	**38.28** | **0.958** (3.58)	**34.74** | **0.932** (2.17)	**32.47** | **0.909** (1.59)	30.81 | 0.885 (1.31)	**29.56** | 0.857 (1.16)
MLFM	37.82 | 0.953 (3.80)	34.55 | 0.930 (2.09)	**32.47** | 0.908 (1.58)	**30.86** | **0.886** (1.32)	29.53 | 0.857 (1.17)
BPFA	37.26 | 0.949 (4.50)	34.22 | 0.927 (2.40)	32.23 | 0.907 (1.72)	30.67 | 0.885 (1.45)	29.51 | **0.859** (1.27)
KSVD	38.05 | 0.956 (7.04)	34.45 | 0.930 (3.11)	32.41 | 0.907 (1.77)	**30.86** | 0.881 (1.16)	29.55 | 0.852 (0.83)
TV(ANISO)	34.17 | 0.936 (—)	29.77 | 0.877 (—)	27.49 | 0.820 (—)	26.00 | 0.770 (—)	25.07 | 0.728 (—)
TV(ISO)	34.18 | 0.936 (—)	29.77 | 0.877 (—)	27.50 | 0.822 (—)	26.01 | 0.773 (—)	25.12 | 0.734 (—)
BASELINE	34.16 | 0.887 (—)	28.14 | 0.724 (—)	24.61 | 0.594 (—)	22.11 | 0.497 (—)	20.18 | 0.422 (—)
GOLDHILL	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$
iMLFM	35.72 | 0.935 (3.22)	32.70 | 0.881 (1.65)	31.12 | **0.838** (1.22)	**30.03** | **0.799** (1.09)	**29.15** | **0.764** (1.03)
MLFM	35.20 | 0.928 (3.20)	32.64 | 0.878 (1.58)	31.15 | 0.837 (1.24)	30.00 | 0.797 (1.09)	29.14 | 0.762 (1.03)
BPFA	34.73 | 0.924 (3.05)	32.17 | 0.875 (1.39)	30.87 | 0.833 (1.30)	29.89 | 0.790 (1.16)	29.09 | 0.762 (1.08)
KSVD	**36.65** | **0.941** (6.78)	**33.25** | **0.885** (2.49)	**31.40** | 0.832 (1.25)	30.01 | 0.787 (0.77)	29.05 | 0.746 (0.53)
TV(ANISO)	34.73 | 0.908 (—)	31.43 | 0.833 (—)	29.74 | 0.776 (—)	28.61 | 0.732 (—)	27.79 | 0.696 (—)
TV(ISO)	34.81 | 0.910 (—)	31.52 | 0.836 (—)	29.83 | 0.781 (—)	28.69 | 0.736 (—)	27.87 | 0.700 (—)
BASELINE	34.16 | 0.897 (—)	28.14 | 0.727 (—)	24.61 | 0.577 (—)	22.11 | 0.461 (—)	20.18 | 0.373 (—)
LENA	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$
iMLFM	37.49 | 0.934 (2.39)	34.99 | **0.905** (1.53)	33.51 | 0.879 (1.25)	32.38 | 0.861 (1.12)	**31.43** | **0.844** (1.06)
MLFM	37.36 | 0.931 (2.29)	35.07 | **0.905** (1.53)	33.58 | 0.880 (1.26)	**32.42** | **0.862** (1.13)	31.40 | 0.843 (1.07)
BPFA	37.41 | 0.932 (2.56)	34.92 | 0.903 (1.73)	33.41 | **0.882** (1.45)	32.31 | 0.861 (1.25)	31.17 | 0.840 (1.13)
KSVD	**38.26** | **0.937** (4.22)	**35.38** | **0.905** (1.66)	**33.68** | 0.879 (0.91)	32.40 | 0.856 (0.61)	31.30 | 0.832 (0.43)
TV(ANISO)	35.92 | 0.917 (—)	32.71 | 0.874 (—)	30.96 | 0.841 (—)	29.84 | 0.816 (—)	28.87 | 0.793 (—)
TV(ISO)	35.95 | 0.917 (—)	32.78 | 0.874 (—)	31.04 | 0.843 (—)	29.93 | 0.818 (—)	28.98 | 0.796 (—)
BASELINE	34.16 | 0.855 (—)	28.14 | 0.646 (—)	24.61 | 0.493 (—)	22.11 | 0.390 (—)	20.18 | 0.317 (—)
PEPPERS	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$
iMLFM	36.47 | 0.935 (2.30)	34.23 | 0.924 (1.46)	33.00 | **0.905** (1.23)	31.94 | 0.884 (1.12)	**31.09** | **0.871** (1.05)
MLFM	35.97 | 0.931 (2.12)	**34.28** | **0.926** (1.46)	**33.01** | **0.905** (1.24)	**31.99** | **0.885** (1.13)	**31.09** | 0.869 (1.07)
BPFA	35.61 | 0.937 (2.33)	34.10 | 0.920 (1.64)	32.45 | 0.904 (1.38)	31.37 | 0.882 (1.23)	30.46 | 0.864 (1.11)
KSVD	**37.72** | **0.949** (4.82)	34.20 | 0.923 (1.73)	32.16 | 0.900 (0.90)	30.80 | 0.877 (0.58)	29.64 | 0.855 (0.42)
TV(ANISO)	35.73 | 0.938 (—)	32.40 | 0.903 (—)	30.44 | 0.872 (—)	29.25 | 0.850 (—)	28.26 | 0.828 (—)
TV(ISO)	35.85 | 0.938 (—)	32.56 | 0.905 (—)	30.59 | 0.875 (—)	29.42 | 0.853 (—)	28.40 | 0.832 (—)
BASELINE	34.16 | 0.855 (—)	28.14 | 0.642 (—)	24.61 | 0.485 (—)	22.11 | 0.382 (—)	20.18 | 0.310 (—)

iterations and the time cost is similar across all models.

We show the mean and variance of the predictive log-likelihood for all models in Figure 3. Since the nonparametric models are not a function of latent feature, we show their performance as a straight line. We observe that the performance for MLFM starts to decrease when K reaches 25 and its best performance is the same as the nonparametric iMLFM, which is not surprising. We also observe that nonparametric model performance is ordered as BPFA $<$ nCRP $<$ iMLFM. We believe that this improvement in performance is a result of the increasing ability to model the relationships between features in this sequence.

We also show some qualitative results for these models. In Figure 2 we show the 0-1 feature assignment matrix for the three nonparametric models. To the left we show the country of origin for the person associated with each row. In the columns we show the most heavily used features and reorder them for a better visualization. Since BPFA assumes all the features are mutually independent, it has a harder time exploiting the natural structure in the data. The nCRP gives a better result since it learns a hierarchy of features spread across countries and continents. However, since the model uses a strict tree structure as shown on the RHS of Figure 2, it may not be flexible enough to uncover all the hierarchical correlations.

On the RHS of Figure 2 we also show the graph learned by iMLFM, where the 13 significantly used features (and the null state, denoted by a dark node) are displayed. The entire graph looks like a tree when organized according to transition probability as we have done, but there are some differences. First, the structure is not a strict hierarchy. For example, there are transitions from the null state to the "leaves." There are also transitions across "subtrees." Thus, we learn a graph structure that approximates a tree, following the structure of the data, but allows for individuals not to adhere strictly to this (in this case because, e.g., there may have been some ancestor from another region). This shows the flexibility of our Markov-structured model.

5.2. Image denoising

We also experiment using scalable inference for an image denoising task, where we would like to recover the original image from an image corrupted by white Gaussian noise. We demonstrate results for four 512×512 images: 'Barbara', 'Goldhill', 'Lena', and 'Peppers' (below in Figure 5). We extract 8×8 patches from the noise-corrupted im-

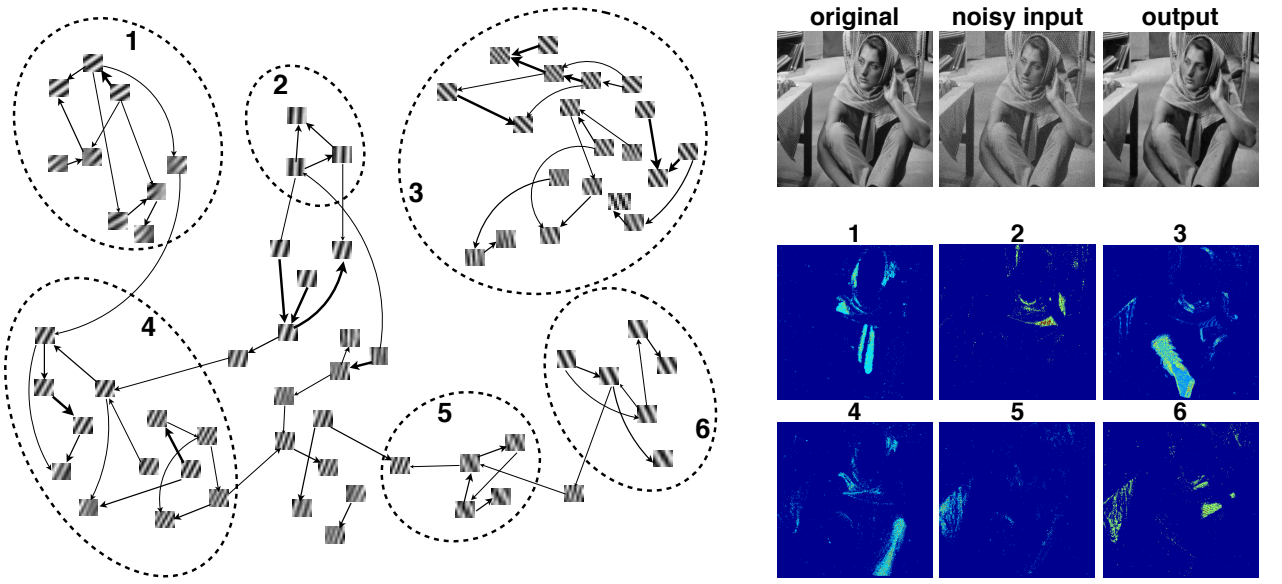

Figure 4. (Left) A similarity-preserving 2-d embedded graph of the learned dictionary elements using their transition probabilities. We show edges about a threshold, and the direction of an edge between two dictionary elements according to the higher transition probability. (Right) Feedback maps for dictionary elements in various regions in the graph on the left.

ages using a single-pixel sliding window to scan the entire image. This gives a total of 255,025 patches.

We compare with scalable BPFA (Sertoglu & Paisley, 2015), the non-probabilistic K-SVD model (Aharon et al., 2006), and isotropic and anisotropic total variation (TV) (Goldstein & Osher, 2009). We set $\eta = 1/255^2$, $\lambda = 1/10$, $\alpha = 1$, $\gamma = 1$, $K = 256$, and online parameters $|C_t| = 1000$, $t_0 = 10$, $\kappa = 0.75$. We truncated iMLFM to 256 states. For all methods, we set σ using the method from (Liu et al., 2013); for TV we found the regularization parameter that resulted in this empirical noise variance. For the stochastic algorithms, we train using 500 iterations, which was enough for convergence; thus the number of patches seen during inference was equivalent to two passes through the entire dataset. To quantify the recovery quality, we show the peak signal-to-noise ratio (PNSR) and the structured similarity (SSIM) performance measures (Wang et al., 2004).

We show the result on various images using different noise standard deviations in Table 1. As a baseline performance measure, we use the original noisy image. We can see that iMLFM often performs better than BPFA and has results comparable to K-SVD. In the images 'Barbara' and 'Peppers', iMLFM performs better than K-SVD. We also show the average number of features occupied by each patch in Table 1. iMLFM is sparser than K-SVD when σ is small, which is a matter of balancing predictive gain and extra cost of growing paths in our greedy sparse coding step. For the running time, MLFM takes about 4 minutes to converge, which is similar to K-SVD and BPFA.

Figure 5. The four images used in our experiments.

Finally, we show some qualitative result on the 'Barbara' image in Figure 4. We show the graph learned by iMLFM using a similarity-preserving 2-d embedding, where edges having probabilities above a threshold are displayed between dictionary elements. Since iMLFM learns a directed graph, we show the direction of edges according to the larger transition probability between two elements. For the ease of visualization, we only show the latent features that contain stripes. On the LHS of Figure 4, we see that the direction of stripes varies, but stripes with similar directions have greater connectivity. The graph also has a global structure as the similarity-preserving embedding shows. For example, the direction of stripes changes from the direction '\' in the right area, to '|' in the center area, and to '/' in the left area.

On the RHS of Figure 4, we display the feedback map of dictionary elements in the six regions defined on the LHS. As we can see, some groups of features give a local feed-

back that has semantic meanings. For example, Region 1 (9 features) can be interpreted as the scarf; Region 3 (20 features) is the right leg; Region 4 (12 features) is the left leg; Region 5 (5 features) is the tablecloth. Above we show the ground truth image, the noisy input and the denoised output for the corresponding experiment.

6. Conclusion

We presented a Markov latent feature model (MLFM) using a simple sequential construction and connected this construction to the requisite Markov property of the stochastic process. The key is through the Markov exchangeability constraint, which allows for a mixing measure to be defined for easy variational inference. This procedure for constructing latent features models allows for feature correlations to be learned from the data, and so in a sense we have presented a "correlated IBP"-type model. We showed two simple examples of a Markov latent feature model, one parametric and one nonparametric, and scaled inference to handle large datasets. Empirical results on a genome analysis task and an image denoising task demonstrates the advantage of correlated feature modeling.

Appendix

6.1. Proof of Proposition 2

First, note the $q(\mathbf{C}_i)$ is a function of $\widehat{\mathbf{Z}}_i$. The objective function in Eq. (7) becomes

$$\mathcal{L}(\mathbf{Z}_{\psi(i)})) = \mathcal{L}_1(\mathbf{Z}_{\psi(i)}) + \mathcal{L}_2(\mathbf{Z}_{\psi(i)}), \tag{9}$$

where

$$\begin{aligned}\mathcal{L}_1(\mathbf{Z}_{\psi(i)}) &= \mathbb{E}_q[\ln p(\mathbf{Z}_{\psi(i)}|\boldsymbol{\theta})] \\ \mathcal{L}_2(\mathbf{Z}_{\psi(i)}) &= \mathbb{E}_q[\ln p(\mathbf{X}_i|\mathbf{C}_i, \widehat{\mathbf{Z}}_i, \mathbf{W})] + f(\widehat{\mathbf{Z}}_i).\end{aligned} \tag{10}$$

Here $f(\widehat{\mathbf{Z}}_i) = \mathbb{E}_q[\ln \frac{p(\mathbf{C}_i)}{q(\mathbf{C}_i)}]$, by marginalizing out $\mathbf{C}_i$. Note that $\mathcal{L}_2(\mathbf{Z}_{\psi(i)})$ can only take finite values, since there are finite configurations for $\widehat{\mathbf{Z}}_i$. We only need to prove that $\mathcal{L}_1(\mathbf{Z}_{\psi(i)})$ cannot always be improved. We have

$$\mathcal{L}_1(\mathbf{Z}_{\psi(i)}) = \sum_{j=\tau(i-1)+1}^{\tau(i)} \mathbb{E}_q[\ln \theta_{Z_{j-1},Z_j}]. \tag{11}$$

Let $\mathcal{L}_1^{(1)} = \sum_{j=\tau(i-1)+1}^{\tau(i)-1} \mathbb{E}_q[\ln \theta_{Z_{j-1},Z_j}]$ be the cost of all transitions execpt for the last one, and $\mathcal{L}_1^{(2)}$ be the rest part of $\mathcal{L}_1$. Note that as the sequence grows, $\mathcal{L}_1^{(1)}$ is monotonically decreasing, since $\theta_{Z_{j-1},Z_j} < 1$. And $\mathcal{L}_1^{(2)}$ (the cost of last transition plus a constant) can only take finite values. Thus, $\mathcal{L}_1$ cannot monotonically increase through greedy search.

6.2. Stochastic Variational Inference

We use stochastic variational inference (SVI) (Hoffman et al., 2013) to scale up MLFM by using stochastic optimization with natural gradients. Suppose N is large, the objective function for $\mathbf{W}$ is

$$\mathcal{L}(\mathbf{W}) = \ln p(\mathbf{W}) + \sum_{i=1}^{N} \mathbb{E}_q[\ln p(\mathbf{X}_i|\mathbf{C}_i, \widehat{\mathbf{Z}}_i, \mathbf{W})]. \tag{12}$$

At iteration t we sample a subset indexed by C_t and set the objective

$$\mathcal{L}_t(\mathbf{W}) = \ln p(\mathbf{W}) + \frac{N}{|C_t|} \sum_{i\in C_t} \mathbb{E}[\ln p(\mathbf{X}_i|\mathbf{C}_i, \widehat{\mathbf{Z}}_i, \mathbf{W})]. \tag{13}$$

Then we can perform an unbiased natural gradient decent for $\mathbf{W}$ as follows:

$$\begin{aligned}\mathbf{B}_t &= \eta\sigma^2 \frac{|C_t|}{N} I + \sum_{i\in C_t} \widetilde{\mathbf{Z}}_i(\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top + \Sigma_i)\widetilde{\mathbf{Z}}_i, \\ \mathbf{W}_t' &= (\textstyle\sum_{i\in C_t} \mathbf{X}_i\boldsymbol{\mu}_i^\top \widetilde{\mathbf{Z}}_i)\mathbf{B}_t^{-1}, \\ \mathbf{W}^{(t+1)} &= (1-\rho_t)\mathbf{W}^{(t)} + \rho_t \mathbf{W}_t'.\end{aligned} \tag{14}$$

Similarly, to update the conditional posterior of $\boldsymbol{\theta}$, we have $q(\boldsymbol{\theta}) = \prod_{k=0}^{K} q(\boldsymbol{\theta}_k)$, where $q(\boldsymbol{\theta}_k) = \mathrm{Dir}(\mathbf{a}_k)$. We update

$$\begin{aligned}\mathbf{a}'_{j,j'} &= \frac{\alpha}{K+1} + \frac{N}{|C_t|} \sum_{i\in C_t} \sum_{i'=\tau(i-1)+1}^{\tau(i)} \mathbb{1}_{\{Z_{i'}=j, Z_{i'+1}=j'\}}, \\ \mathbf{a}_{j,j'}^{(t+1)} &= (1-\rho_t)\mathbf{a}_{j,j'}^{(t)} + \rho_t \mathbf{a}'_{j,j'},\end{aligned} \tag{15}$$

where $\rho_t = (t+t_0)^{-\kappa}$, and $\kappa \in (0.5, 1]$.

Acknowledgements. This work was supported in part by Laboratory Directed Research and Development (LDRD) funding from Lawrence Livermore National Laboratory under contract B616449.

References

Aharon, M., Elad, M., and Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

Bacallado, S., Favaro, S., Trippa, L., et al. Bayesian nonparametric analysis of reversible Markov chains. *The Annals of Statistics*, 41(2):870–896, 2013.

Blei, D. M., Griffiths, T. L., and Jordan, M. I. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

Broderick, T., Pitman, J., Jordan, M. I., et al. Feature allocations, probability functions, and paintboxes. *Bayesian Analysis*, 8(4):801–836, 2013.

Diaconis, P. and Freedman, D. De Finetti's theorem for Markov chains. *The Annals of Probability*, 8(1):115–130, 1980.

Diaconis, P. and Rolles, S. W. Bayesian analysis for reversible Markov chains. *The Annals of Statistics*, pp. 1270–1292, 2006.

Ghahramani, Z. and Hinton, G. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, 1996.

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

Goldstein, T. and Osher, S. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

Griffiths, T. and Ghahramani, Z. The Indian buffet process: An introduction and review. *The Journal of Machine Learning Research*, 12:1185–1224, 2011.

Heaukulani, C. and Roy, D. M. The combinatorial structure of Beta negative Binomial processes. *arXiv preprint arXiv:1401.0062*, 2013.

Heaukulani, C., Knowles, D., and Ghahramani, Z. Beta diffusion trees. In *International Conference on Machine Learning (ICML)*, pp. 1809–1817, 2014.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Johnson, M. J. and Wilsky, A. S. Stochastic variational inference for Bayesian time series models. In *International Conference on Machine Learning (ICML)*, 2014.

Knowles, D., Ghahramani, Z., and Palla, K. A reversible infinite HMM using normalised random measures. In *International Conference on Machine Learning (ICML)*, pp. 1998–2006, 2014.

Liang, P., Petrov, S., Jordan, M. I., and Klein, D. The infinite PCFG using hierarchical Dirichlet processes. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, 2007.

Liu, X., Tanaka, M., and Okutomi, M. Single-image noise level estimation for blind denoising. *IEEE Transactions on Image Processing*, 22(12):5226–5237, 2013.

Norris, J. R. *Markov Chains*. Cambridge University Press, 1998.

Paisley, J. and Carin, L. Nonparametric factor analysis with Beta process priors. In *International Conference on Machine Learning (ICML)*, pp. 777–784, 2009.

Paisley, J., Wang, C., Blei, D. M., and Jordan, M. I. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, 2015.

Rosenberg, N. A., Pritchard, J. K., Weber, J. L., Cann, H. M., Kidd, K. K., Zhivotovsky, L. A., and Feldman, M. W. Genetic structure of human populations. *science*, 298(5602):2381–2385, 2002.

Sertoglu, S. and Paisley, J. Scalable Bayesian nonparametric dictionary learning. In *European Signal Processing Conference (EUSIPCO)*, 2015.

Shah, A., Knowles, D. A., and Ghahramani, Z. An empirical study of stochastic variational algorithms for the Beta Bernoulli process. In *International Conference on Machine Learning (ICML)*, 2015.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Thibaux, R. and Jordan, M. I. Hierarchical Beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 564–571, 2007.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Zhang, A. and Paisley, J. Markov mixed membership models. In *International Conference on Machine Learning (ICML)*, 2015.

Zhang, A., Gultekin, S., and Paisley, J. Stochastic variational inference for the HDP-HMM. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Zhou, M., Padilla, O., and Scott, J. Priors for random count matrices derived from a family of negative binomial processes. *Journal of the American Statistical Association*, (to appear), 2015.