Landmark Dependent Hierarchical Beta Process

Anonymous Author(s)
Affiliation
Address
email

Abstract

A landmark-dependent hierarchical beta process is developed for nonparametric Bayesian dictionary learning, for data with associated covariates. The landmark locations are learned, to which the data are linked through normalized kernels. A dictionary-usage probability vector is associated with each landmark, encouraging data with similar covariates to be represented in terms of similar dictionary atoms. We consider two distinct applications: (i) denoising of an image corrupted by a superposition of white Gaussian and spiky noise, and (ii) video background/foreground modeling. The same framework achieves state-of-the-art results on both applications, without parameter tuning.

1 Introduction

There has been significant recent interest in dictionary learning and sparse coding, with applications in denoising, interpolation, feature extraction and classification [1–8]. In the nonparametric Bayesian framework, it has been shown that dictionary learning and sparse coding can be naturally formulated under a sparse factor analysis (SFA) model [9], using the Indian buffet process (IBP) [10–12] or the closely related beta-Bernoulli process [13–15] as the sparseness-promoting prior [16]. The factor loading and factor score correspond to the dictionary and sparse codes, respectively.

These nonparametric Bayesian priors usually assume exchangeability between samples, and there has been a fruitful line of research on removing this assumption in the Dirichlet and Chinese restaurant processes, by introducing covariate-dependence [17–20]. Similar efforts have been devoted to the IBP. The phylogenetic IBP imposes prior knowledge on inter-sample relationships via a tree structure [21]. A dependent IBP (dIBP) model has been introduced recently, with a hierarchical Gaussian process (GP) used to account for covariate dependence [22]. When considering large-scale applications, the utilization of a GP may be computationally expensive. Following these efforts, a dependent hierarchical beta process (dHBP) has been recently introduced to regularize dictionary learning with covariate-dependence [23], replacing the basic beta process (BP) as the sparseness-promoting prior. Using an $N \times N$ random-walk matrix summarizing the relationship between the N samples in covariate space, dHBP substantially outperforms BP and demonstrates state-of-the-art performance on image reconstruction applications. Despite the success of dHBP, it has unfavorable scaling with N, which we wish to ameliorate in this paper.

Based on the hierarchical beta process (HBP) [13], we address nonparametric Bayesian dictionary learning and sparse coding for data that are endowed with an associated covariate. We explore the idea that data nearby in covariate space are likely to be represented in terms of a similar subset of dictionary atoms. Hence we employ "landmarks" in covariate space to guide the dictionary usage probabilities, with the landmarks learned along with the dictionary atoms. We refer to the proposed model as landmark-dependent HBP (Landmark-dHBP), whose limiting cases are BP and dHBP for landmark sizes $J = 1$ and $J = N$, respectively. With $J \ll N$, Landmark-dHBP has comparable computational complexity to BP, and is considerably more efficient than dHBP for large-scale learning and out-of-sample prediction.

We develop theoretical properties of Landmark-dHBP, summarize an efficient inference algorithm, and apply it to challenging learning tasks. Specifically, we demonstrate state-of-the-art results for dictionary learning employed for (i) denoising of an image corrupted by a superposition of white Gaussian and spiky noise, and (ii) video background/foreground modeling.

2 Review of motivating models

Following [13], a beta process (BP) is a positive random measure on a space Ω, where a BP draw is denoted $B \sim \text{BP}(c, B_0)$; c is a positive function over Ω, B_0 is a fixed measure on Ω (called the base measure), and in the following we assume c is a positive constant. A BP draw can be expressed as $B = \sum_{k=1}^{\infty} p_k \delta_{\omega_k}$, where $\omega_k \sim B_0/B_0(\Omega)$ is the kth atom, and each p_k is drawn i.i.d. from a degenerate beta distribution with parameter c. We draw the atom usage vector for the ith sample, $i = 1, \ldots, N$, from a Bernoulli process as $X_i \sim \text{BeP}(B)$. The generalization of the beta-Bernoulli process to the Indian buffet process, and a hierarchical generalization of BP (HBP) are discussed in detail in [13].

The BP construction above is exchangeable. Building on the HBP [13], a dependent HBP (dHBP) was recently proposed to introduce covariant dependence, demonstrating significant improvements over BP in image interpolation and denoising applications. The model can be express as

$$B_i = \sum_{j=1}^{N} a_{ij} B_j^*,\ B_j^* \sim \text{BP}(c_1, B), B \sim \text{BP}(c_0, B_0),\ a_{ij} = \mathcal{K}(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j) / \sum_{j'=1}^{N} \mathcal{K}(\boldsymbol{\ell}_i, \boldsymbol{\ell}_{j'}) \qquad (1)$$

where c_0 and c_1 are constants, a_{ij} is an element of the random-walk matrix $\mathbf{A} \in \Re^{N \times N}$, whose ith row $\boldsymbol{a}_i = [a_{i1}, \ldots, a_{iN}]$ sums to one, $\boldsymbol{\ell}_i \in \mathbb{R}^{\mathcal{L}}$ is a covariate associated with the ith sample, and the kernel $\mathcal{K}(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j)$ monotonically decreases towards 0 with increasing $\|\boldsymbol{\ell}_i - \boldsymbol{\ell}_j\|$. As shown in [23], for any measurable subset S, the dependent random measures B_i and $B_{i'}$ constituted as above satisfy

$$\text{corr}\{B_i(S), B_{i'}(S)\} = \frac{< \boldsymbol{a}_i, \boldsymbol{a}_{i'} >}{\|\boldsymbol{a}_i\| \cdot \|\boldsymbol{a}_{i'}\|}. \qquad (2)$$

When constructing the random-walk matrix, [23] used a neighborhood constraint on the kernel that $\mathcal{K}(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j) \neq 0$ if and only if $j \in \mathcal{Q}_i$, where $\mathcal{Q}_i$ includes the indexes of the L nearest covariates of $\boldsymbol{\ell}_i$ from $\{\boldsymbol{\ell}_j\}_{j=1,N}$. This construction, enforcing that each row of $\mathbf{A}$ has only L nonzero elements, directly links B_j^* to $\{B_i\}_{\{i:\{j \in \mathcal{Q}_i\}\}}$. Since given B_i, X_i is independently drawn as $X_i \sim \text{BeP}(B_i)$, this construction makes it feasible to infer the posterior distribution of B_j^* based on $\sum_{i:\{j \in \mathcal{Q}_i\}} X_i$.

3 Landmarks and kernels

The salutary characteristics of (2) are undermined by several additional attributes. First, the complexity of the model grows with the number of data samples N, even if many of them provide redundant information. Second, inference on a new sample i' is complicated by the fact that one must retain all training data to compute the weights $\{a_{i'j}\}$ when subsequently using the model. We seek to remove these deficiencies of the dHBP model, while retaining its desirable characteristics, particulary the covariate-dependent removal of exchangeability. Toward this end, we introduce the concept of "landmarks" in the covariate space.

Consider the J landmarks $\{\tilde{\boldsymbol{\ell}}_j\}_{j=1,J}$ to be learned, where each $\tilde{\boldsymbol{\ell}}_j \in \mathbb{R}^{\mathcal{L}}$. These landmarks are meant to capture the support of the set of covariates $\{\boldsymbol{\ell}_i\}_{i=1,N}$, serving as reference points to guide the covariate-dependent usage of atoms $\{\omega_k\}$. We are typically interested in $J \ll N$. For general covariate $\boldsymbol{\ell}$, the landmark-dependent probability measure for atom usage is expressed as

$$B_{\boldsymbol{\ell}} = \sum_{j=1}^{J} a_{\boldsymbol{\ell} j} B_j^*, B_j^* \sim \text{BP}(c_1, B), B \sim \text{BP}(c_0, B_0),\ a_{\boldsymbol{\ell} j} = \mathcal{K}(\boldsymbol{\ell}, \tilde{\boldsymbol{\ell}}_j) / \sum_{j'=1}^{J} \mathcal{K}(\boldsymbol{\ell}, \tilde{\boldsymbol{\ell}}_{j'}) \qquad (3)$$

where $\mathcal{K}(\boldsymbol{\ell}_i, \tilde{\boldsymbol{\ell}}_j)$ is a kernel as discussed above. Note that $B_{\boldsymbol{\ell}}$ varies smoothly with the covariate $\boldsymbol{\ell}$. To complete the model specification, the landmarks are assumed drawn from a probability measure H in $\mathbb{R}^{\mathcal{L}}$, as $\tilde{\boldsymbol{\ell}}_j \sim H$. A natural and simple approach is to densely select $\widetilde{N}$ positions $\{\boldsymbol{\ell}_i\}_{i=1,\widetilde{N}}$ in the covariate space as potential landmarks, and assume $H = \sum_{i=1}^{\widetilde{N}} \theta_i \delta_{\boldsymbol{\ell}_i}$ with $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{\widetilde{N}}]^T$ drawn from a Dirichlet distribution. In this way the model selects a subset of the $\widetilde{N}$ covariates as landmarks. The kernel is constructed as

$$\mathcal{K}(\boldsymbol{\ell}_i, \tilde{\boldsymbol{\ell}}_j) = \delta\big(j \in \mathcal{Q}_i\big) \exp(-\|\boldsymbol{\ell}_i - \tilde{\boldsymbol{\ell}}_j\|_2/\sigma) \tag{4}$$

where the kernel width σ is a constant and $\mathcal{Q}_i$ includes the indexes of the L nearest landmarks of $\boldsymbol{\ell}_i$ from $\{\tilde{\boldsymbol{\ell}}_j\}_{j=1,J}$. We now have a random walk matrix $\mathbf{A} \in \Re^{N\times J}$, each row of which $\boldsymbol{a}_i \in \Re^J$ has L nonzero components and sums to one. We are typically interested in $J \ll N$, with dHBP and BP recovered as two limiting cases as J increases to N and decreases to one, respectively.

3.1 Properties

Consider a fixed set of landmarks $\{\tilde{\boldsymbol{\ell}}_j\}$. It directly follows (2) that draws from the landmark-dependent measure $B_{\boldsymbol{\ell}}$ in (3) satisfy corr$\{B_{\boldsymbol{\ell}}(S), B_{\boldsymbol{\ell}'}(S)\} = < \boldsymbol{a}_{\boldsymbol{\ell}}, \boldsymbol{a}_{\boldsymbol{\ell}'} > /(\|\boldsymbol{a}_{\boldsymbol{\ell}}\| \cdot \|\boldsymbol{a}_{\boldsymbol{\ell}'}\|)$, where $\boldsymbol{a}_{\boldsymbol{\ell}} = [a_{\boldsymbol{\ell}1}, \ldots, a_{\boldsymbol{\ell}J}]$. This is true for *any* fixed set of landmarks, $\{\tilde{\boldsymbol{\ell}}_j\}$, which may not necessarily be well matched to the data. When using the proposed prior in a model to fit data, the likelihood function encourages the posterior distribution on landmark positions to be well matched to the data, as discussed below when considering specific applications.

To address further model properties, conditioned on fixed landmark positions $\{\tilde{\boldsymbol{\ell}}_j\}$, we consider the marginal properties by which atoms are drawn. For sample i we draw a set of atoms represented by X_i, where $X_i = \sum_{k=1}^{\infty} b_{ik}\delta_{\omega_k}$; $X_i \sim$ BeP$(B_{\boldsymbol{\ell}_i})$, with BeP$(\cdot)$ representing a Bernoulli process. If $B_{\boldsymbol{\ell}_i} = \sum_{k=1}^{\infty} \pi_{\boldsymbol{\ell}_i k}\delta_{\omega_k}$, then $b_{ik} \sim$ Bernoulli$(\pi_{\boldsymbol{\ell}_i k})$. Recall [13] that if $X_1 \sim$ BeP(B) with $B \sim$ BP(c, B_0), then marginalizing out B we have $X_1 \sim$ BeP(B_0). Assuming B_0 is non-atomic, this implies that X_1 is composed of $m_1 \sim$ Poisson$(B_0(\Omega))$ atoms, each drawn i.i.d. from $B_0/B_0(\Omega)$. Assuming X_1 is so drawn, we have $B|X_1 \sim$ BP$(c+1, \frac{c}{c+1}B_0 + \frac{1}{c+1}X_1)$, and again marginalizing out B we have $X_2|X_1 \sim$ BeP$(\frac{c}{c+1}B_0 + \frac{1}{c+1}X_1)$, yielding $X_2 = X_2^{(n)} + X_2^{(e)}$, with $X_2^{(e)} \sim$ BeP$(\frac{1}{c+1}X_1)$ and $X_2^{(n)} \sim$ BeP$(\frac{c}{c+1}B_0)$. Expression $X_2^{(e)}$ selects from among previously *existing* atoms (as defined by X_1), and $X_2^{(n)}$ selects a set of m_2 *new* atoms from B_0.

For fixed landmarks $\{\tilde{\boldsymbol{\ell}}_j\}$, the generative process may be expressed as

$$X_i \sim \text{BeP}(B^*_{z_i}) \ , \ z_i|\boldsymbol{\ell}_i \sim \sum_{j=1}^{J} a_{\boldsymbol{\ell}_i j}\delta_j \ , \ B^*_j \sim \text{BP}(c_1, B) \ , \ B \sim \text{BP}(c_0, B_0) \tag{5}$$

where z_i is a latent indicator variable. If the generative process is followed sequentially to constitute $\{X_i\}_{i=1,N}$, one may show that with $z_{N+1}|\boldsymbol{\ell}_{N+1} \sim \sum_{j=1}^{J} a_{\boldsymbol{\ell}_{N+1}j}\delta_j$

$$X_{N+1} = X_{N+1}^{(n)} + X_{N+1}^{(e)} \tag{6}$$

where $X_{N+1}^{(n)} \sim$ BeP$(\frac{c_1}{c_1+n_{z_{N+1}}}V_N)$ and $X_{N+1}^{(e)} \sim$ BeP$(\frac{n_{z_{N+1}}}{n_{z_{N+1}}+c_1}\sum_{i=1}^{N}\delta(z_i = z_{N+1})X_i)$ with $V_N = \frac{c_0}{c_0+N}B_0 + \frac{N}{N+c_0}\sum_{i=1}^{N} X_i^{(n)}$ and $n_j = \sum_{i=1}^{N}\delta(z_i = j)$.

Each landmark has an associated "local" buffet. The probability of which landmark (and associated local buffet) sample i with covariate $\boldsymbol{\ell}_i$ visits is dictated by its proximity in covariate space to all J such buffets, computed via $\{\boldsymbol{a}_{\boldsymbol{\ell}_i j}\}_{j=1,J}$; the closer the landmark, the more likely it is visited. When sample $N+1$ visits its corresponding local buffet (landmark), it may sample dishes (atoms) that already exist at that buffet, $\sum_{i=1}^{N}\delta(z_i = z_{N+1})X_i$, with the probability of selecting from among these dishes dictated by their popularity among the previous $n_{z_{N+1}}$ customers to that local buffet. In addition, customer $N+1$ may select new dishes from a "global" buffet $\sum_{i=1}^{N} X_{ni}$ shared among all covariate-dependent local buffets; this global buffet accounts for all dishes across all N previous samples. Hence, X_{N+1} has two contributions, with $X_{N+1}^{(n)}$ coming from the global buffet, and $X_{N+1}^{(e)}$ from existing dishes at the local buffet associated with landmark $\tilde{\boldsymbol{\ell}}_{z_{N+1}}$. When drawing from the global buffet, a new dish may also be drawn from B_0; however, when $N \gg c_0$ it is unlikely that such new dishes will be selected. Further, as n_j becomes large relative to c_1, any subsequent customer to the jth local buffet is unlikely to select dishes off the global buffet.

4 Dictionary learning with Landmark-dHBP

4.1 Beta process factor analysis

As shown in Fig. 1, a beta process sparse factor analysis (BPFA) model [16] can be constructed as

$$\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i), \gamma_\epsilon^{-1}\mathbf{I}_P), \ \ \boldsymbol{d}_k \sim \mathcal{N}(0, P^{-1}\mathbf{I}_P), \ \ \boldsymbol{s}_i \sim \mathcal{N}(0, \gamma_s^{-1}\mathbf{I}_K) \tag{7}$$

$$z_{ik} \sim \text{Bernoulli}(\pi_k), \ \ \pi_k \sim \text{Beta}(c\eta, c(1-\eta)) \tag{8}$$

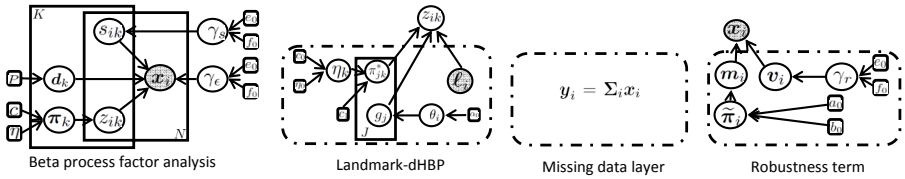

Figure 1: Graphical model for beta process factor analysis, Landmark-dHBP, the missing data layer and robustness term.

where $\boldsymbol{x}_i$ is the ith sample, $\odot$ represents the Hadamard product, $\mathbf{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K] \in \mathbb{R}^{P \times K}$ is the dictionary, $\boldsymbol{s}_i = [s_{i1}, \ldots, s_{iK}]^T$, $\boldsymbol{z}_i = [z_{i1}, \ldots, z_{iK}]^T$, $s_{ik} \in \mathbb{R}$, $z_{ik} = X_i(\boldsymbol{d}_k) \in \{0, 1\}$ indicates whether the kth atom is *active* within sample i, and $\pi_k = B(\boldsymbol{d}_k)$ is the probability for the kth atom to be selected. Gamma hyper-priors are placed on both γ_ϵ and γ_s. Additional hierarchical components shown in Fig. 1 and explained below – including the covariate-dependence term, missing-data layer and robustness term – do not change the basic structure of the model. Therefore, the model inference and complexity stays about the same, and these additional components can be independently turned on or off based on the requirements of specific applications.

4.2 Landmark-dHBP sparse factor analysis

Assuming we have data and associated covariates $\{\boldsymbol{x}_i, \boldsymbol{\ell}_i\}_{i=1,N}$, when employing the landmark-based construction in (3), (8) generalizes as

$$z_{ik} \sim \text{Bernoulli}(\pi_{ik}), \quad \pi_{ik} = \sum_{j=1}^{J} a_{ij} \pi^*_{jk}, \tag{9}$$

$$\pi^*_{jk} \sim \text{Beta}(c_1\eta_k, c_1(1-\eta_k)), \quad \eta_k \sim \text{Beta}(c_0\eta_0, c_0(1-\eta_0)) \tag{10}$$

where $z_{ik} = X_{\boldsymbol{\ell}_i}(\boldsymbol{d}_k)$, $\pi_{ik} = B_{\boldsymbol{\ell}_i}(\boldsymbol{d}_k)$, $\pi^*_{jk} = B^*_{\tilde{\boldsymbol{\ell}}_j}(\boldsymbol{d}_k)$, $\eta_k = B(\boldsymbol{d}_k)$ and a_{ij} is constructed via the landmarks as in (3). In the applications, we use all the observed covariates as potential landmarks and learn the landmark positions with

$$\tilde{\boldsymbol{\ell}}_j = \boldsymbol{\ell}_{g_j}, \quad g_j \sim \sum_{i=1}^{N} \theta_i \delta_i \ , \ \boldsymbol{\theta} \sim \text{Dir}(\alpha_0, \ldots, \alpha_0). \tag{11}$$

In one application considered below, the objective is to denoise an image, potentially in the presence of non-Gaussian noise. In this case the data $\{\boldsymbol{x}_i\}$ correspond to pixels from image patches, and the associated covariates $\{\boldsymbol{\ell}_i\}$ locate patch positions within the image. For applications in which data reside on a manifold (or approximate manifold), and we do not have spatial covariates as above, we may define covariates as sample locations relative to each other. Specifically, we can define

$$\boldsymbol{\ell}_i = \boldsymbol{x}_i / \|\boldsymbol{x}_i\|_2 \tag{12}$$

and we have $\|\boldsymbol{\ell}_i - \boldsymbol{\ell}_{i'}\|_2 = \sqrt{2 - 2\cos(\boldsymbol{x}_i, \boldsymbol{x}_{i'})}$, where the cosine distance $\cos(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \boldsymbol{x}_i^T \boldsymbol{x}_{i'} / (\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_{i'}\|_2)$ is widely used to measure similarity between vectors. We will utilize (12) for dictionary learning when spatial covariates in an image are unavailable.

After the dictionary and landmark locations are learned on the training data, the jth landmark is encoded with a dictionary usage probability vector as $\boldsymbol{\pi}^*_j = [\pi^*_{j1}, \ldots, \pi^*_{jK}]^T$. Although we do not consider such in the applications below, this framework may readily be employed to encode a new sample, without having to return to the training data (as required in [23]). For the sparse coding of a new sample $\boldsymbol{x}_{i'}$, we can calculate its kernel distances to the J landmarks and calculate $\boldsymbol{\pi}_{i'}$ as

$$a_{i'j} = \frac{\mathcal{K}(\boldsymbol{\ell}_{i'}, \tilde{\boldsymbol{\ell}}_j)}{\sum_{j'=1}^{J} \mathcal{K}(\boldsymbol{\ell}_{i'}, \tilde{\boldsymbol{\ell}}_{j'})}, \ \ \pi_{i'k} = \sum_{j=1}^{J} a_{i'j} \pi^*_{jk}. \tag{13}$$

With $\boldsymbol{\pi}_{i'}$ and the learned dictionary $\mathbf{D}$, we can infer the landmark-dependent sparse codes of $\boldsymbol{x}_{i'}$.

4.3 Handling missing data and outliers

For data with missing components, we can add an additional layer to the model, as $\boldsymbol{y}_i = \boldsymbol{\Sigma}_i \boldsymbol{x}_i$, where $\boldsymbol{\Sigma}_i$ is the sampling matrix (rows of $\boldsymbol{\Sigma}_i$ are all zeros except for a single one, and the rows are

orthogonal); $\boldsymbol{\Sigma}_i$ indicates which components of $\boldsymbol{x}_i$ are observed, and we only use the observed data $\{\boldsymbol{y}_i\}_{i=1,N}$ to infer the latent parameters for missing data estimation.

For data with outliers, motivated by recent increased interest in robust PCA (RPCA) [24–26], we assume that in addition to the Gaussian noise $\boldsymbol{\epsilon}_i$, there are additive sparse spiky data components as

$$\boldsymbol{x}_i = \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i) + \boldsymbol{\epsilon}_i + \boldsymbol{v}_i \odot \boldsymbol{m}_i \tag{14}$$

where $\boldsymbol{v}_i = [v_1, \ldots, v_P]^T$ and $\boldsymbol{m}_i = [m_1, \ldots, m_P]^T$, $v_{ip} \in \mathbb{R}$ is the weight and m_{ip} is the binary spiky indicator. A beta-Bernoulli sparseness-promoting prior is constituted on $\boldsymbol{m}_i$ as $m_{ip} \sim$ Bernoulli$(\tilde{\pi}_{ip})$ and $\tilde{\pi}_{ip} \sim$ Beta(a_0, b_0). We impose $\boldsymbol{v}_i \sim \mathcal{N}(0, \gamma_v^{-1}\mathbf{I}_P)$ and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \gamma_\epsilon^{-1}\mathbf{I}_P)$, with gamma hyperpriors on γ_v and γ_ϵ. After performing analysis with this model, the underlying data are estimated as $\hat{\boldsymbol{x}}_i = \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i)$, (ideally) free of outliers and noise.

4.4 Brief discussion of inference

The inference is performed using MCMC analysis, with slice sampling for η_k, Metropolis-Hastings (M-H) for π^*_{jk} and g_j and Gibbs sampling for all the other parameters. In practice we have found that the MCMC inference mixes well, yielding robust results for a wide class of sophisticated applications, as discussed below. Space limitations prohibit listing the update equations here, but they are provided as Supplemental Material.

5 Example results

The model parameters are here set as $a_0 = 1$, $b_0 = 100$, $c_0 = 10$, $c_1 = 1$, $\eta_0 = 0.5$, and the gamma hyper-priors are $e_0 = f_0 = 10^{-6}$; these parameters were not optimized, and many related settings yield similar results. Note that the landmark size J and the number of closest landmarks L are two parameters selected based on a compromise between computational complexity and the desired level of covariant dependence. In general, as J increases, more finely calculated landmark-dependent sparseness properties are obtained, but the computational complexity increases at a fast rate, and the number of latent parameters to estimate also rapidly increases, which might lead to slower mixing and hence slower convergence. Let L_0 denote the prior belief that a local region formed by a set of L_0 nearest samples share similar features, we set $L = \max\{\lfloor L_0 \frac{J}{N} \rceil, 2\}$, where $\lfloor \cdot \rceil$ represents the closest integer, and L is lower bounded by 2 to enforce the dependance in case J is too small. All the experiments are implemented with non-optimized Matlab, and the computational time is reported based on a 2.67 GHz PC.

5.1 Local latent feature discovery and spiky noise removal

We consider denoising an image corrupted by a superposition of both sparse spiky noise and white Guassian noise (WGN). We consider the 256×256 House image, with similar results found on many different images (omitted for brevity). We extract all 8×8 overlapping patches and use their spatial locations as covariates. When the patches are spatially proximate they tend to be similar; by contrast, the spiky noise associated with nearby patches are less likely to share common features. However, if one examines all patches, independent of their location (as in an exchangeable model), it is possible for widely separated patches to have similar spiky anomalies; in fact, for large images with many patches, it is likely. Hence, we anticipate that the spatial covariates, which localize dictionary learning, will yield significant advantages for this problem ($i.e.$, good results are anticipated for dHBP and Landmark-dHBP, but poor results are likely for BP and related non-Bayesian approaches [26], because the latter employ the exchangeability assumption).

Due to the spatial contiguity observed in natural images, we set $L_0 = 29$ based on the assumption that a patch and its 28 spatial neighbors (within the radius of 3) are similar. Our objective is to separate out unusual components (spiky noise) within local spatial regions. The dictionary size is set as $K = 256$ and not all the atoms are used, based on the posterior usage of dictionary atoms (not shown, for brevity). With the same robustness term in (14) to model the sparse spiky noise, we apply BP, dHBP, and Landmark-dHBP to discover latent local features.

We first consider *in situ* feature learning, in which no *a priori* training images are employed, and all the latent parameters (including the dictionary) are inferred solely based on the corrupted data under test. As shown in Figs. 2(b) and (g), BP learns dictionary atoms containing sparse spiky noise, which are *not* rare when observed across the entire image, leading to the worst denoising performance

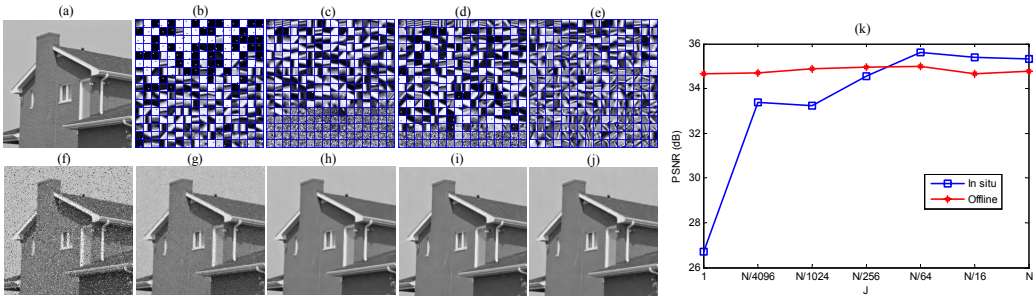

Figure 2: Denoisng results of BP, dHBP, Landmark-dHBP on "House" image using in-situ or offline learning. 10% pixels are corrupted by spiky noise situated uniformly at random, whose amplitudes are uniformly distributed between -255 and 255. Zero-mean WGN with standard deviation 10 is also added to the image. (a) the original image; (f) the noisy image; (b)-(d) show dictionaries learned *in situ*, and (g)-(i) show the denoised images for BP, dHBP and Landmark-dHBP with $J = \lfloor N/64 \rceil$, respectively; (e) shows a dictionary learned *off line* with Landmark-dHBP, and (j) show the denoised images for Landmark-dHBP with $J = \lfloor N/64 \rceil$; (k) shows the denoising results as a function of the landmark size J for both *in situ* and *off line* learning.

(PSNR of 26.71 dB); this is a result of the exchangeability assumption. The dHBP and Landmark-dHBP models use the locations of the patch centers as covariates, imposing spatial-dependence and spatial locality. As shown in Figs. 2(c) and (h), dHBP successfully regularizes dictionary learning and sparse coding, leading to uncontaminated dictionary atoms and much better denoising results (35.32 dB). The Landmark-dHBP learns the positions of landmarks, in addition to learning the dictionary, and as shown in Figs. 2(d) and (i), with $J = \lfloor N/64 \rceil$, where $N = (256 - 8 + 1)^2 = 62001$, it learns localized dictionary atoms and has the best denoising results (although it learns more spiky contaminated atoms than dHBP, it discovers the most local structures). If we set J equal to $\lfloor N/4096 \rceil$, $\lfloor N/1024 \rceil$, $\lfloor N/256 \rceil$, $\lfloor N/64 \rceil$, and $\lfloor N/16 \rceil$, each MCMC iteration takes about 20, 27, 31, 49 and 95 seconds, and it learns respectively 14, 61, 232, 801 and 2564 unique landmark positions. As shown by the blue curve in Fig. 2(k), as the landmark size increases, the performance generally improves until it saturates. The results reported here are based on 2000 burn-in and 500 collection samples (results averaged across collection samples), but we find that even 250 MCMC iterations can produce satisfactory denoising results, whose PSNRs are about 1 dB worse than those reported here.

We also considered using Landmark-dHBP to learn a dictionary offline, based on 3×10^4 patches of size 8×8, extracted at random from 300 images in the Berkeley image segmentation dataset (similar offline dictionary learning has been employed elsewhere, for example in [27]). In this case we do not have spatial covariates, because the patches come from many different images. We therefore for this application use (12) as covariates; by using (12), we impose that patches that are similar in form are likely to be represented by similar dictionary atoms. The offline-learned dictionary with $K = 256$ is shown in Fig. 2(e). This dictionary is then fixed within BP, dHBP and Landmark-dHBP for executing the denoising of test images. With this offline-learned dictionary, on the test image dHBP and Landmark-dHBP again employ covariates defined by the spatial locations of the patches in the image to be denoised, as was done above for online learning. Hence, in this application covariates are employed in two distinct forms: (i) Landmark-dHBP employs (12) to learn the dictionary offline, and (ii) this learned dictionary is employed on the noisy test image, and for denoising spatial locations of the patches are used as covariates by dHBP and Landmark-dHBP.

Figure 2(j) shows the denoising results based on offline dictionary learning, for the best choice of J, and the red curve in Fig. 2(k) shows PSNR values of the offline approach, for all J considered. From these results we infer that offline learning of the dictionary provides reasonable results. However, it only recovers major structure in the image, and fails to recover detailed structure unique to the image under test, such as the brick texture on the wall of the house. We also tried using a larger size of training data and a larger size of dictionary, but the performance gains were insignificant. This is not surprising, since an offline-learned dictionary tends to capture general behavior across a wide range of randomly selected images, and therefore it may miss detailed local structure specific to a test image. By contrast, in the *in situ* dictionary feature learning, with covariate-dependence introduced via Landmark-dHBP and dHBP, the model is able to recover detailed local structures from the corrupted data, leading to the best results. This phenomemon was also observed on many other images, especially for these with complicated local structure.

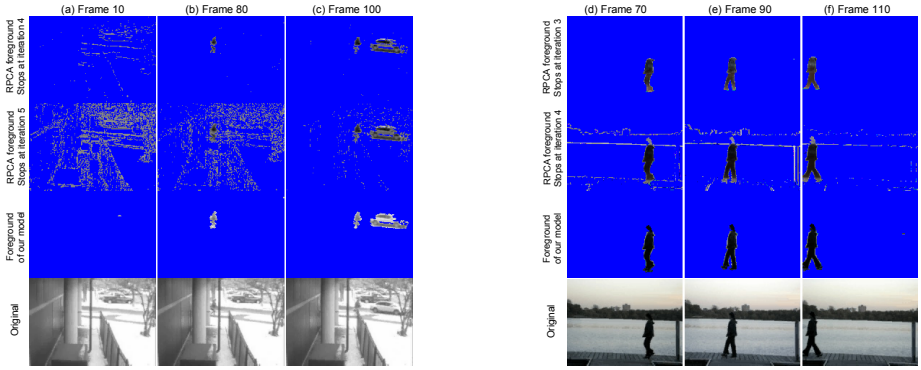

Figure 3: Comparison of the extracted foreground between RPCA and the proposed model, on a gray-scale and a RGB videos with constant irregular camera motion. Detected non-foreground pixels are displayed in blue, and detected foreground pixels of RPCA are scale normalized for visualization. (a)-(c), (d)-(f): Each column is a frame and its rows from top to bottom are the RPCA extracted foreground, using the best two stopping points of ALM [26] found with experiments, the foreground extracted by our model and the original video, respectively.

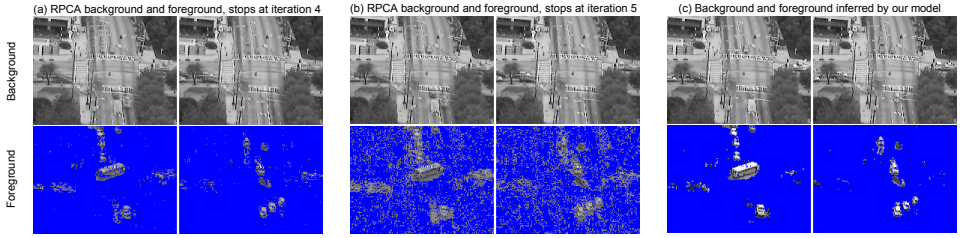

Figure 4: Comparison of the extracted background and foreground between RPCA and the proposed model. (a)-(c) Detected background and foreground of RPCA and for the proposed model; in each sub-figure, the left and right columns show frames 50 and 150, respectively.

As a comparison to a non-Bayesian approach to this denoising problem, we also considered the augmented Lagrange multiplier (ALM) [26] method, a state-of-the-art implementation of RPCA, designed for removal of spiky noise like that considered here. In this case, with the $N = 62001$ patches in the image, ALM was here applied to a matrix of size $\mathbb{R}^{64\times 62001}$. ALM proved ineffective at removing the noise, most likely because of a violation of the low-rank assumption; additionally, ALM does not employ covariates, and in that sense it is like BP (assumes exchangeability).

5.2 Video background/foreground modeling

Recent research on robust PCA (RPCA) has resulted in new ways to analyze video. RPCA [24] constitutes a video matrix $\mathbf{X} \in \mathbb{R}^{P\times N}$, with N frames, each of dimension P (after "unwrapping" the image into a vector). RPCA employs the decomposition $\mathbf{X} = \mathbf{L} + \mathbf{V}$, where the background $\mathbf{L}$ is modeled as low rank and the foreground $\mathbf{V}$ is modeled as sparse. When implementing RPCA one typically must pre-specify the stopping criteria based on the noise variance or the "true" rank of the background, making careful parameter tuning unavoidable.

Under the nonparametric Bayesian dictionary learning framework, we propose to replace (14) with

$$\boldsymbol{x}_i = \boldsymbol{l}_i \odot (1 - \boldsymbol{m}_i) + \boldsymbol{v}_i \odot \boldsymbol{m}_i + \boldsymbol{\epsilon}_i \tag{15}$$

to model a video frame, where $\boldsymbol{l}_i = \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i)$ is the modeled background, while $\boldsymbol{l}_i \odot (1 - \boldsymbol{m}_i)$, and $\boldsymbol{v}_i \odot \boldsymbol{m}_i$ are models of the actually observed background and foreground, respectively. Further, $m_{ip} \in \{0, 1\}$ is a binary background/foreground indicator, with $m_{ip} \sim \text{Bernoulli}(\tilde{\pi}_{ip})$. The noise is now explicitly modeled, and its variance is inferred (RPCA requires knowledge of the noise variance); specifically, we assume $\boldsymbol{\epsilon}_i$ has components drawn i.i.d. from a Gaussian with associated precision that is inferred, using a gamma prior. Within the video analysis, we employ covariates in two ways. First, landmark-dHBP is employed with time (frame index) as the covariate, encouraging proximate frames to employ similar dictionary atoms to model the background (this part of the model controls $\boldsymbol{z}_i$ in (15)). Secondly, we employ pixel locations as covariates, and use Landmark-dHBP (more precisely, an *analogous* construction without explicit dictionary atoms) to model the variation

of $\tilde{\pi}_{ip}$ as a function of pixel location; in this case we do not impose a vector of probabilities with each landmark, but only one probability, associated with $\tilde{\pi}_{ip}$ above. This latter use of landmarks controls $\boldsymbol{m}_i$, encouraging spatially and temporally contiguous foreground objects. All videos from the below analysis can be viewed online at *http://www.youtube.com/user/bayesianvideo*.

As a comparison, we consider the ALM algorithm [26], a state-of-the-art implementation of RPCA, and search the two best stopping points through experiments, with which we compare to the results of our model (ALM is often sensitive to its stopping point, so we tune that to provide its best results). It is anticipated that results from the proposed model will be significantly better than ALM because the former imposes more structure: frame consistency on the background and pixel consistency on the foreground. Additionally, the proposed method has the advantage of *learning* the noise variance, and therefore there are no stopping-point issues with the associated inference.

The first example has 101 frames of gray-scale video 176×120 pixels, and the second has 120 frames of RGB video with 160×120 pixels; both videos are characterized by irregular camera motion and real noise. The proposed model required 10 and 25 seconds per MCMC iteration, respectively, for these two examples. Results are shown in Fig. 3, demonstrating that the proposed dual-landmark model is robust to the non-stationary background, and accurately detects the full body of the moving objects, even when parts of them are not easily visible due to low contrast relative to the background. For reasons anticipated above, RPCA is much more sensitive to camera motion, and it is sensitive to the stopping criteria. In both examples, when the iterations are stopped early enough to produce a reasonable foreground estimation, it estimates the background with a rank too small to reveal background variations. Whereas if the iterations are stopped later, it produces reasonable background estimation, but the foreground is severely corrupted with noise.

The third video example has 250 gray-scale frames, each frame 144×192, corresponding to a traffic-surveillance video. This video is fairly complicated since the background is time-varying, *e.g.*, there are major changes when the traffic light signal switches, and the foreground objects appear in a large variety of intensities, sizes, shapes and quantities. As shown in Fig. 4, the proposed model unambiguously captures background transitions caused by traffic light switching, and captures well the moving objects, including walking persons and moving vehicles. Pixel intensity fluctuations caused by waving trees are modeled as non-stationary noise, affecting neither the background nor the foreground. By contrast, RPCA yields an overly smoothed background estimate, and it usually breaks slowly moving objects into both background and foreground. In addition, if it does not stop early enough (in its iterative solution), the detected foreground is severely corrupted by noise.

6 Conclusions

A landmark-dependent hierarchical beta process is developed and applied for nonparametric Bayesian dictionary learning and sparse coding. The model infers a relatively low-dimensional set of landmarks in covariate space, and each landmark has an associated probability vector, controlling dictionary usage. The probability of dictionary usage varies smoothly as a function of covariates, as manifested by a kernel that relates the landmark covariates to the covariates of the data of interest. Results have been presented for a series of sophisticated image-processing and video-analysis applications, motivated by recent research on robust PCA [24–26]. In these applications the covariates have taken different forms. Specifically, we have considered the problem of denoising an image characterized by a combination of Gaussian and spiky noise. When the dictionary atoms are learned *in situ*, with no training data, the covariates corresponded to the location of patches in the image. When we learned a dictionary offline, based on a set of training images, the covariates were related to the relative similarity of the patches, imposing that patches with similar structure should be characterized in terms of similar dictionary usage. The proposed model yielded state-of-the-art performance on the denoising application, significantly better than BP, and much more efficient computationally than dHBP [23]. We also presented state-of-the-art results on separating background/foreground in video. In this case the covariates took two different forms, simultaneously within the same analysis. Specifically, to impose consistency of dictionary usage in representing the background, from frame to frame in the video, the covariate corresponded to the frame index (time). Additionally, landmarks were also learned in the spatial pixel space, imposing that foreground pixels (objects) should be spatially and temporally contiguous. The flexibility of the model to address these diverse and complicated inference tasks, and the quality of the results, demonstrate the promise for exploiting covariates in the context of Bayesian modeling, and particularly when performing dictionary learning.

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations. *IEEE Trans. Signal Process.*, 2006.

[2] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 2006.

[3] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[4] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *submitted to IEEE TIP*, 2010.

[5] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.

[6] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 2009.

[7] Y. Jiang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[8] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[9] M. West. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, 7:723–732, 2003.

[10] T.L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *NIPS*, 2005.

[11] D. Knowles and Z. Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *Independent Component Analysis and Signal Separation*, 2007.

[12] P. Rai and H. Daumé. The infinite hierarchical factor regression model. In *NIPS*, 2008.

[13] R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. In *AISTATS*, 2007.

[14] J. Paisley and L. Carin. Nonparametric factor analysis with beta process priors. In *ICML*, 2009.

[15] Y. W. Teh and D. Gorur. Indian buffet processes with power-law behavior. In *NIPS*, 2009.

[16] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric Bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.

[17] J. Grifin and M. Steel. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 2006.

[18] D.B. Dunson, N.S. Pillai, and J.-H. Park. Bayesian density regression. *J. Royal. Statist. Soc B.*, 2007.

[19] D.B. Dunson and J.-H. Park. Kernel stick-breaking processes. *Biometrika*, 2008.

[20] D.M. Blei and P. Frazier. Distance dependent Chinese restaurant processes. In *ICML*, 2010.

[21] K. Miller, T. Griffiths, and M. I. Jordan. The phylogenetic Indian buffet process: A non-exchangeable nonparametric prior for latent features. In *UAI*, 2008.

[22] S. Williamson, P. Orbanz, and Z. Ghahramani. Dependent Indian buffet processes. In *AISTATS*, 2010.

[23] M. Zhou, H. Yang, G. Sapiro, D.B. Dunson, and L. Carin. Dependent hierarchical beta process for image interpolation and denoising. In *AISTATS*, 2011.

[24] E.J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *to apear in Journal of the ACM*, 2011.

[25] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *NIPS*, 2009.

[26] Z. Lin, M. Chen, and Y. Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *UIUC Technical Report*, 2009.

[27] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010.