Image Understanding in a Nonparametric Bayesian Framework

Soumya Ghosh
sghosh@cs.brown.edu

Abstract

We explore recently proposed nonparametric Bayesian statistical models of image partitions. These models are attractive because they adapt to images of different complexity, successfully modeling uncertainty in size, shape, and structure of human segmentations of natural scenes. We improve upon them in a number of key ways to achieve performance comparable to state-of-the-art methods. Our first major contribution is a novel discrete search based posterior inference algorithm which, compared to previous approaches, is significantly more robust and accurate. We then present a low rank version of the spatially dependent Pitman-Yor processes model, critical for efficient inference. Furthermore, we show how the Gaussian process covariance functions underlying the proposed models can be calibrated to accurately match the statistics of human segmentations. Finally, we present accurate segmentations of complex scenes as well as multiple hypothesized image partitions (capturing the inherent uncertainty in human scene interpretations) produced by our method.

1 Introduction

Image understanding, or interpreting images by locating and characterizing their content, is arguably the holy grail of computer vision. A general image understanding system must flexibly deal with "stuff" (materials) and "things" (objects) [1]. Forsyth et al. [8], define stuff as "a homogeneous or repetitive pattern of fine-scale properties, but no specific or distinctive spatial extent or shape" while a thing is defined as "an object with specific shape and size". For instance, foliage, sky and gravel are examples of stuff, while cars, tigers and boats are examples of objects (Figure 1). Traditionally, statistical models have dealt with either stuff (under the umbrella of image segmentation) or things (object detectors) [21] but rarely both. Recently however, some progress has been made in leveraging one model to better learn the other. Typically, object models for a fixed number of object categories are specified and learnt from training data. These are then used to detect potential objects in an image. These predictions are then combined in a coherent fashion using "stuff" models. For instance, Heitz et al. [9] use "stuff" based clusters (segments) to prune away false positives from the predictions of a sliding window based car ("thing") detector.

Figure 1: Stuff and Things

The success of such models depend crucially on accurate "stuff" modeling, i.e., on producing accurate image segmentations, which is the primary focus of this work.

Image segmentation deals with the problem of partitioning images into "homogeneous" chunks based on their appearance, location and possibly even semantic content. It has seen a large amount of research in the computer vision community over the past several decades [3, 7, 17]. However, in spite of the intense amount of work, segmentation remains a largely unsolved problem. Part of the reason behind disappointing results is the fact that existing segmentation algorithms tend to be semi-automatic at best. They often come endowed with a host of tunable parameters, which need to be adjusted for each image until the produced segmentations look "reasonable". Furthermore, some popular techniques (e.g., [17]) have implicit biases which encourage the segments to be of roughly equal size. This is in sharp contrast to the segments produced by humans, which tend to span a wide range of sizes even in a single image. Sudderth and Jordan [18] have recently put forth spatially dependent Pitman-Yor process hierarchical mixture models which make a first attempt at addressing many of these issues. In this paper, we describe various improvements necessary to make this approach competitive with state-of-the-art methods.

Our first major contribution involves a new posterior inference algorithm. In [18] the authors propose a mean field based variational inference algorithm. Mean field methods are known to be highly susceptible to local optima. As a result, there is reason to believe that the promising results of [18] can be further improved with a better inference technique. In particular, we combine a discrete stochastic search to make large moves in the space of image partitions, with an accurate higher-order variational approximation (based on expectation propagation) to marginalize high-dimensional continuous latent variables. Our results do indeed show improved accuracy and robustness to initialization.

Next, we present a novel low rank representation of the model presented in [18]. Such a representation significantly reduces the computational burden of Bayesian inference, allowing for a useful image segmentation algorithm. Our next contribution lies in replacing various manually tuned parameters (in [18]) with ones estimated quantitatively from human segmentations.

Also, note that because we employ a nonparametric model we do not need to specify the number of segments observed in each image. In fact we infer a posterior distributions over segmentations of varying structure and resolution. We provide interesting examples of multiple modes of this posterior distribution. Lastly, we demonstrate that our overall performance is both quantitatively and qualitatively competitive with state-of-the-art methods.

2 Nonparametric Bayesian Segmentation

In this section, we fist review various nonparametric Bayesian models proposed in the literature for modeling image partitions. In Sec. 2.4, we then propose a model which exploits the low-rank representation of the Gaussian distributions underlying our model. This is essential for the computational tractability of our later algorithms.

2.1 Image Representation

We begin by first dividing each image into roughly 1,000 *superpixels* [15] using the normalized cuts spectral clustering algorithm [17]. The color of each superpixel is described using a histogram of HSV color values with $W_c = 120$ bins. We choose a non-regular quantization to more coarsely group low saturation values. Similarly, the texture of each superpixel is modeled via a local $W_t = 128$ bin texton histogram [12], using quantized band-pass filter responses. Superpixel i is then represented by histograms $x_i = (x_i^t, x_i^c)$ indicating its texture x_i^t and color x_i^c.

2.2 Pitman-Yor Mixture Models

Natural scenes contain widely varying numbers of objects of varying sizes. Not surprisingly, human segmentations of natural scenes also consist of segments of widely varying sizes. It has been observed that histograms over segment areas [11] and contour lengths [14] are well explained by power law distributions. Previous work [18] has shown that such power law distributions in natural images are well modeled via the Pitman-Yor process [13].

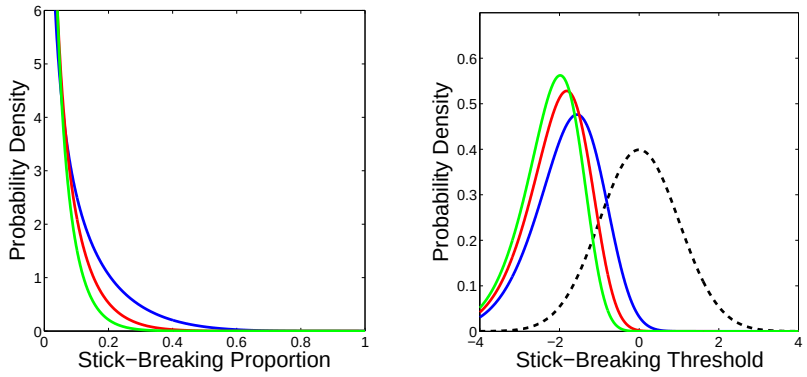

Figure 2: Learned Pitman-Yor prior model for image partitions, where $\boldsymbol{\pi} \sim \mathrm{GEM}(0.6, 3.0)$. *Left:* Beta distributions from which stick proportions w_k are sampled for $k = 1$ (blue), $k = 10$ (red), $k = 20$ (green). *Right:* Corresponding distributions on thresholds for an equivalent generative model employing zero mean, unit variance Gaussians (dashed black).

Here, we consider the stick breaking representation of the Pitman-Yor (PY) process. Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \ldots)$, $\sum_{k=1}^{\infty} \pi_k = 1$, denote an infinite *partition* of a unit area region (in our case, an image). The Pitman-Yor process defines a prior distribution on this partition via the following *stick-breaking* construction:

$$\pi_k = w_k \prod_{\ell=1}^{k-1} (1 - w_\ell) = w_k \left(1 - \sum_{\ell=1}^{k-1} \pi_\ell\right)$$
$$w_k \sim \mathrm{Beta}(1 - \alpha_a, \alpha_b + k\alpha_a) \tag{1}$$

This distribution, denoted by $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha_a, \alpha_b)$, is defined by two hyperparameters satisfying $0 \leq \alpha_a < 1$, $\alpha_b > -\alpha_a$. When $\alpha_a = 0$, we recover a *Dirichlet process* (DP) with concentration parameter α_b. For the DP, $\mathbb{E}[\pi_k]$ decreases exponentially with k; for the PY, it instead has a heavier-tailed, polynomial decay rate $\mathbb{E}[\pi_k] \propto k^{-1/\alpha_a}$.

For image segmentation, each index k is associated with a different segment or region with its own appearance models $\theta_k = (\theta_k^t, \theta_k^c)$ parameterized by multinomial distributions on the W_t texture and W_c color bins, respectively. Each superpixel i then independently selects a region $z_i \sim \mathrm{Mult}(\boldsymbol{\pi})$, and a set of quantized color and texture responses according to

$$p\big(x_i^t, x_i^c \mid z_i, \boldsymbol{\theta}\big) = \mathrm{Mult}\big(x_i^t \mid \theta_{z_i}^t, M_i\big)\, \mathrm{Mult}(x_i^c \mid \theta_{z_i}^c, M_i)\,. \tag{2}$$

Note that conditioned on the region assignment z_i, the color and texture features for each of the M_i pixels within superpixel i are sampled independently. The appearance feature channels provide weak cues for grouping superpixels into regions. Since, the model doesn't enforce any spatial neighborhood cues, it is referred to as the "bag of features"(*BOF*) model (Figure 3).

2.3 Spatially Dependent Pitman-Yor Process Mixture Models

Here, we review the approach of Sudderth and Jordan [18] which extends the BOF model with spatial grouping cues. The model is a generalization of earlier work on generalized spatial Dirichlet process models [6] and combines ideas from Bayesian nonparametrics with ideas from layered models of image sequences [22], and level set representations for segment boundaries [5].

We begin by elucidating the analogy between PY processes and layered image models. Consider the PY stick-breaking representation of Eq. (1). If we sample a random variable z_i such that $z_i \sim \mathrm{Mult}(\boldsymbol{\pi})$ where $\pi_k = w_k \prod_{\ell=1}^{k-1}(1 - w_\ell)$, it immediately follows that $w_k = \mathbb{P}[z_i = k \mid z_i \neq k-1, \ldots, 1]$. The stick-breaking proportion w_k is thus the *conditional* probability of choosing segment k, given that segments with indexes $\ell < k$ have been rejected. If we further interpret the ordered PY regions as a sequence of layers, z_i can be sampled by proceeding through the layers in order, flipping biased coins (with probabilities w_k) until a layer is chosen.

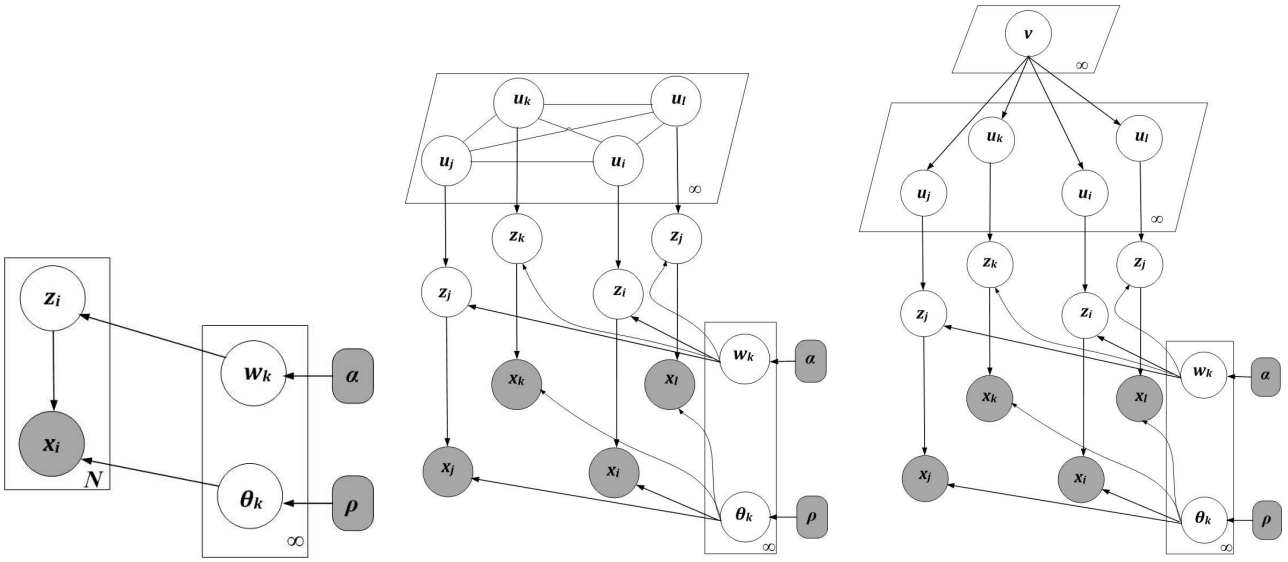

Figure 3: Generative models of image partitions. ***From left to right***. Bag of features model, full rank model, low rank model.

Given this, the probability of assignment to subsequent layers is zero; they are effectively *occluded* by the chosen "foreground" layer.

The spatially dependent Pitman-Yor process of [18] preserves this PY construction,while adding spatial dependence among superpixels, via thresholded zero mean *Gaussian processes* (GPs) $\mathbf{u}_k$,

$$z_i = \min\left\{k \mid u_{ki} < \Phi^{-1}(w_k)\right\}, \quad u_{ki} \sim \mathcal{N}(0,1)\,. \tag{3}$$

Here, $u_{ki} \perp u_{\ell i}$ for $k \neq \ell$ and $\Phi(u)$ is the standard normal *cumulative distribution function* (CDF). Let $\delta_k = \Phi^{-1}(w_k)$ denote the threshold for layer k. Since $\Phi(u_i)$ is uniformly distributed on $[0,1]$, we have

$$\mathbb{P}[z_i = 1] = \mathbb{P}[u_{1i} < \delta_1] = \mathbb{P}[\Phi(u_{1i}) < w_1] = w_1 \tag{4}$$

$$\mathbb{P}[z_i = 2] = \mathbb{P}[u_{1i} > \delta_1]\,\mathbb{P}[u_{2i} < \delta_2] = (1 - w_1)w_2 \tag{5}$$

and so on. The extent of each layer is determined via the region on which a real-valued function lies below some threshold, akin to level set methods. If the GPs have identity covariance functions, we recover the basic PY mixture model. More general covariances can be used to encode the prior probability that each feature pair occupies the same segment; developing methods for learning these probabilities is a major contribution of this paper.

The power law prior on segment sizes is retained by transforming priors on stick proportions $w_k \sim \mathrm{Beta}(1-\alpha_a, \alpha_b + k\alpha_a)$ into corresponding randomly distributed thresholds $\delta_k = \Phi^{-1}(w_k)$:

$$p(\delta_k \mid \alpha) = \mathcal{N}(\delta_k \mid 0,1) \cdot \mathrm{Beta}(\Phi(\delta_k) \mid 1-\alpha_a, \alpha_b + k\alpha_a) \tag{6}$$

Fig. 2 illustrates the threshold distributions corresponding to a PY stick-breaking prior learned from human segmentations in Sec. 4. Figure 3 displays the graphical model corresponding to the spatially dependent Pitman-Yor process Mixture model.

2.4 Low-Rank Covariance Representations

In the preceding generative model, the layer support functions $u_k \sim \mathcal{N}(0, \Sigma)$ are samples from some Gaussian distribution over the N pixels. Analogously to factor analysis models, we can instead employ a low-rank representation based on $D \leq N$ dimensions. Sampling $v_k \sim \mathcal{N}(0, I_D)$, we then set $u_k = A\boldsymbol{v}_k + \epsilon_k$, where A in a N-by-D matrix and $\epsilon_k \sim \mathcal{N}(0, \Psi)$, and Ψ is a diagonal matrix chosen to ensure that $\Sigma = AA^T + \Psi$ has unit diagonal. Figure 3 displays the graphical model to the generative process described in this section.

3 Inference

In this section we provide a detailed description of the inference algorithm for the low rank model. The proposed inference scheme lies in the family of Maximization Expectation (ME) [23] techniques. In contrast to the popular Expectation Maximization techniques, we marginalize out the model parameters and maximize over the latent variables. From figure 3, we observe that our latent variables correspond to segment assignments of super-pixels. Thus, any configuration of these variables defines a partition of the image. Our strategy is to search over the space of these image partitions with the goal of maximizing the posterior marginal likelihood

$$\hat{z} = \underset{z}{argmax}\ p\left(z \mid x, \eta\right) \tag{7}$$

where η represents the set of hyper-parameters governing the model. It is worth noting that since different partitions will have different numbers of segments, we are in fact searching over models of varying complexities as in traditional model selection techniques.

Thus, the proposed inference scheme has to first evaluate the posterior marginal likelihood for a given image partition z. It then has to modify the image partition in an interesting fashion to generate a new partition z' and compute the posterior marginal for z', accepting z' if $p(z' \mid x, \eta) > p(z \mid x, \eta)$. This process is repeated until convergence. In what follows, we first describe the innovations required for evaluating the posterior marginal, followed by the discrete search responsible for generating new partitions from a given partition.

3.1 Posterior Marginal Computation

In our model (*Figure 3*) $p\left(z \mid x, \eta\right)$ factorizes conveniently as

$$\begin{aligned} p\left(z \mid x, \eta\right) &\propto p\left(x \mid z, \rho\right) p\left(z \mid \alpha, A, \Psi\right) &(8)\\ &\propto p\left(z \mid \alpha, A, \Psi\right) \int_{\Theta} p\left(x \mid z, \Theta\right) p\left(\Theta \mid \rho\right) d\Theta &(9)\end{aligned}$$

for the spatial models, where $p\left(z \mid \alpha, A, \Psi\right)$ can be further expanded as follows

$$\begin{aligned} p(z|\alpha, A, \Psi) &= \int_{u_{1\ldots K(z)}} \int_{\delta} p(z|\delta, u_{1\ldots K(z)}) p(u_{1\ldots K(z)}|AA^T + \Psi) p(\delta|\alpha) du_{1\ldots K(z)} d\delta &(10)\\ &= \prod_{k=1}^{K(z)} \int_{u_k} \int_{\delta_k} p\left(z \mid \delta_k, u_k\right) p\left(u_k \mid AA^T + \Psi\right) p\left(\delta_k \mid \alpha\right) du_k d\delta_k &(11)\end{aligned}$$

where $K(z)$ is the number of layers in a given partition. Note that this quantity is a function of the partition z. From here on we denote $K(z)$ by just K, to simplify our notation. Unfortunately, these integrals do not have a closed form solution. Significant innovations are required to evaluate them. For the BOF model z depends only on α and the prior simplifies to $p(z|\alpha)$.

3.1.1 Likelihood Computation

The likelihood computation involves evaluating the independent color and texture integrals

$$\int_{\Theta} p(x \mid z, \Theta) p(\Theta \mid \rho) d\Theta = \int_{\theta^c} p(x^c \mid z, \theta^c) p(\theta^c \mid \rho^c) d\theta^c \int_{\theta^t} p(x^t \mid z, \theta^t) p(\theta^t \mid \rho^t) d\theta^t \tag{12}$$

which is a standard Multinomial-Dirichlet integral, with a closed form solution given by

$$\int_{\Theta} p(x \mid z, \Theta) p(\Theta \mid \rho) d\Theta = \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x_k^{(c)})} \frac{\Delta(\rho^t)}{\Delta(\rho^t + x_k^{(t)})} \tag{13}$$

A detailed derivation is provided in the appendix.

3.1.2 Bag Of Features Prior Evaluation

The prior for the BOF model is just

$$p(z \mid \alpha) = \prod_{k=1}^{K} \int_{w_k} p(z \mid w_k) p(w_k \mid \alpha) dw_k \tag{14}$$

For K segments or regions. The above integral has a closed form solution, which follows from the generalized Chinese restaurant process (*CRP*) representation of the Pitman-Yor process.

$$p(z \mid \alpha) = \alpha_a^K \, \frac{\Gamma(\alpha_b/\alpha_a + K)\,\Gamma(\alpha_b)}{\Gamma(\alpha_b/\alpha_a)\Gamma(N + \alpha_a)} \left(\prod_{k=1}^{K} \frac{\Gamma(M_k - \alpha_a)}{\Gamma(1 - \alpha_a)} \right) \tag{15}$$

Since both the likelihood and the prior for the BOF model can be evaluated in closed form, our search based inference for this model is highly efficient.

3.1.3 Thresholded Gaussian Process prior evaluation

Unfortunately, no closed form solution exists for evaluating the spatial prior. Substantial innovations are required to evaluate the integrals in Equation 11. Note that these integrals can be evaluated independently for each layer. In the following analysis, it is implied that we are dealing with the k^{th} layer allowing us to simplify our notation by dropping the dependence on k. From Figure 3 the per-layer integral can be expressed as follows

$$p(z \mid \alpha, A, \Psi) = \int_u \int_\delta p(z \mid \delta, u)\; p(u \mid AA^T + \Psi)\; p(\delta \mid \alpha) d\delta du = \int_u \int_\delta \int_v p(u, v, \delta, z \mid \alpha) dv d\delta du \tag{16}$$

$$p(u, v, \delta, z \mid \alpha) = p(v)\; p(\delta \mid \alpha) \prod_{n=1}^{N} p(u_n \mid v)\; p(z_n \mid u_n, \delta) \tag{17}$$

Our strategy here is to approximate the joint distribution $p(u, v, \delta, z \mid \alpha)$ with an easy to deal with approximate distribution $q(u, v, \delta, z \mid \alpha)$ and to compute the marginal likelihood $q(z \mid \alpha)$ as an approximation to the true marginal $p(z \mid \alpha)$. Specifically, we choose the approximating distribution to be a Gaussian, and use expectation propagation (EP) to estimate the "closest" such Gaussian.

In the proposed model, z_n is a discrete random variable which takes values in the range $\{1 \ldots K\}$. We now introduce a auxiliary binary random variable t_n, whose value is deterministically related to z_n

$$t_n = \begin{cases} +1 & \text{if } z_n = k \implies u_n < \delta \\ -1 & \text{if } z_n > k \implies u_n > \delta \end{cases} \tag{18}$$

Note that super-pixels with $z_n < k$ have already been assigned to a preceding layer. The corresponding likelihoods are uninformative for the k^{th} layer and are marginalized out before inferring the latent Gaussian function for the k^{th} layer and can be effectively ignored. For each layer we are thus inferring latent Gaussian functions corresponding to a binary classification, with the two class labels being $z_n = k$ or $(t_n = +1)$ and $z_n > k$ or $(t_n = -1)$. Let us now consider the posterior distribution:

$$p(u, v, \delta \mid z, \alpha) = \frac{1}{Z} p(v)\; p(\delta \mid \alpha) \prod_{n=1}^{N} p(u_n \mid v)\; p(z_n \mid u_n, \delta) \tag{19}$$

Equivalently,

$$p(u, v, \delta \mid t, \alpha) = \frac{1}{Z} p(v)\; p(\delta \mid \alpha) \prod_{n=1}^{N} p(u_n \mid v)\; p(t_n \mid u_n, \delta) \tag{20}$$

$$p(u, v, \delta \mid t, \alpha) = \frac{1}{Z} p(v)\; p(\delta \mid \alpha) \prod_{n=1}^{N} p(u_n \mid v) \mathcal{I}(t_n(\delta - u_n) > 0) \tag{21}$$

$$p(u, v, \delta \mid z, \alpha) = \frac{1}{Z} N(v \mid 0, I)\; p(\delta|\alpha) \prod_{n=1}^{N} N(u_n \mid a_n^T \boldsymbol{v}, \psi_n) \mathcal{I}(t_n(\delta - u_n) > 0) \tag{22}$$

where Z is the appropriate normalization constant and $\mathcal{I}$ is an indicator function. At this stage it is worth noting that although we have a binary GP classification problem, it is distinct from the traditional binary GPC presented in the literature. Our problem is complicated by the presence of an additional random variable δ (*the random threshold*) in addition to the random variables (u_n) corresponding to the latent GP functions seen in standard instances of GPC. Furthermore, the prior distribution on δ is a non standard distribution, requiring numerical approximations during EP.

We approximate the likelihood factors $\mathcal{I}(t_n(\delta - u_n) < 0)$ and the threshold prior factor $p(\delta \mid \alpha)$ with un-normalized Gaussians $\tilde{Z}_n \tilde{Z}_{\delta n} \mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) \mathcal{N}(\delta \mid \tilde{\mu}_{\delta n}, \tilde{\sigma}_{\delta n}^2)$ and $\tilde{Z}_p \mathcal{N}(\delta \mid \tilde{\mu}_p, \tilde{\sigma}_p^2)$respectively. The approximate posterior is thus itself a Gaussian distribution

$$q(u, v, \delta \mid z, \alpha) = \frac{1}{Z_{EP}} \mathcal{N}(v \mid 0, I) \mathcal{N}(\delta \mid \tilde{\mu}_p, \tilde{\sigma}_p^2) \prod_{n=1}^{N} \mathcal{N}(u_n \mid a_n^T v, \psi_n) \mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) \mathcal{N}(\delta \mid \tilde{\mu}_{\delta n}, \tilde{\sigma}_{\delta n}^2) \tag{23}$$

where we have absorbed the normalization constants of the un-normalized Gaussians in Z_{EP}. EP can now be run to progressively refine our approximate posterior until convergence.

3.1.4 Posterior Marginal Computation

Finally, we approximate $p(z \mid \alpha)$ with

$$q(z \mid \alpha, A, \Psi) = Z_{EP} = \tilde{Z}_p \prod_n \tilde{Z}_n \int \int \int N(v|0, I) N(\delta \mid \tilde{\mu}_p, \tilde{\sigma}_p^2) \prod_{n=1}^{N} N(u_n \mid a_n^T v, \psi_n) N(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) N(\delta \mid \tilde{\mu}_{\delta n}, \tilde{\sigma}_{\delta n}^2) dv du d\delta \tag{24}$$

We now have all the tools to evaluate $p(z|x, \eta)$ and climb the log posterior $log\ p(z|x, \eta)$ surface. However, note that our likelihood is defined over pixels, while the prior is defined over super-pixels. To balance the prior and likelihood terms, we rescale the log posterior as follows

$$log\ p(z \mid x, \eta) = \frac{1}{\bar{m}}\ log\ p(x \mid z, \rho) + log\ q(z \mid \alpha, A, \Psi) \tag{25}$$

where $\bar{m}$ is the average number of pixels per super-pixel, and climb the rescaled log posterior surface. A more principled approach to likelihood rescaling involves modeling within super-pixel dependencies instead of treating pixels within a super-pixel independently. This is a direction we plan to explore further in future work.

3.1.5 Low Rank Inference

First we note that EP progressively updates the approximate posterior $q(z|\alpha, A, \Psi)$ to be closer to the true posterior. When the approximating family is Gaussian, as is the case here, this amounts to a rank one update of the approximate posterior distribution's precision matrix. Moreover, at least one such rank one update needs to be carried out for each intractable factor in the model.

Observe that the full rank model (Figure 3), has $N + 1$ intractable factors, N likelihood terms one for each super-pixel and one δ term. Updating the posterior for each factor via a rank one update is an $\mathrm{O}(N^2)$ operation. Looping through all intractable factors is an $\mathrm{O}(N^3)$ operation. Evaluating the spatial prior and the posterior marginal $p(z|x, \eta)$ once is thus an $\mathrm{O}(cN^3)$ operation, where c is a constant proportional to the product of number of layers and number of EP iterations. Since, we need to compute the posterior marginal numerous times (in the discrete search phase), we find that for typical settings of $N \approx 1000$, this cost is prohibitively high.

Figure 3 which displays our lower rank model also has $N + 1$ intractable factors. Crucially though, we can compute the moments of the intractable likelihood factors from the moments of v as $E[u_n] = a_n^T E[v]$ and $cov[u_n] = a_n^T cov[v] a_n + \psi_n$. This observation frees us from maintaining an explicit posterior over the N dimensional quantity u. Instead, requiring us to only maintain and update the posterior over the D dimensional quantity v. Thus the cost of evaluating the posterior in the low rank model is $\mathrm{O}(cND^2)$. By setting $D < N$ we can extract significant computational speedups, making the overall search based inference tractable.

3.2 Discrete Tabu Search

We explore the distribution over image segmentations using discrete tabu search. The search performs hill climbing on the posterior probability surface and explores high probability regions of the

segmentation space. This is similar in spirit to MCMC techniques, but has the advantage that it is considerable easier to incorporate flexible search moves. This flexibility allows for robust inference, avoiding local optima problems.

Search moves which change the state of a collection of random variables are referred to as global, while those which change one random variable at a time are local. Our algorithm uses a combination of global and local moves. Given a segmentation we choose from one of the following moves, for a fixed number of iterations, with a new segmentation being accepted if it has a higher posterior probability.

1. Merge - Merge two layers. The segments to be merged are sampled from a uniform distribution over the segments in the current segmentation. Furthermore, we maintain a tabu list of merges, which were proposed but not accepted, to avoid revisiting previously rejected proposals.
2. Split - A layer is split into two. The split move works, by randomly picking a super-pixel in the segment to be split. Next we sample a second super-pixel with probability proportional to its distance(in likelihood space) from the former super-pixel. All other super-pixels are assigned to one of the two selected super-pixel depending on their respective distances from either super-pixel. Note that there are many possible split moves for any segment and it is hence infeasible to maintain a tabu list.
3. Split Connected Components - We also employ another split move, which as the name suggests, splits disconnected components of a segment.
4. Swap - This move is only used with the spatial model. In the spatial model, the relative ordering of segments is explicitly modeled, and partitions with different ordering of segments have different posterior probabilities. The swap move reorders the depth of two segments in the current partition. Here, we again maintain a list of tabu moves.
5. Shift - This is a local move which iterates over all the super-pixels in the image and assigns each super-pixel to a segment which maximizes the posterior probability. The purpose of the shift move is to refine the partitions proposed by the other moves. To understand the working of the shift move observe that for any super-pixel n we have:

$$p(z|x,\eta) \quad \propto \quad p(z_n|z_{-n},\alpha,A,\Psi)p(z_{-n}|\alpha,K)\int_\Theta p(x|z,\Theta)p(\Theta|\rho)d\Theta \tag{26}$$

where z_{-n} refers to random variables corresponding to all but the n^{th} super-pixel. Further, observe that assigning z_n to $\hat{z}_n$ where

$$\hat{z}_n \quad = \quad \underset{z_n}{argmax}\ p(z_n|z_{-n},\alpha,A,\Psi)\int_\Theta p(x|z,\Theta)p(\Theta|\rho)d\Theta \tag{27}$$

$$\approx \quad \underset{z_n}{argmax}\ q(z_n|z_{-n},\alpha,A,\Psi)\int_\Theta p(x|z,\Theta)p(\Theta|\rho)d\Theta \tag{28}$$

takes us a step in the direction of $\hat{z}$ The shift move loops through all super-pixels in an image and assigns each super-pixel n to the corresponding $\hat{z}_n$.

3.3 Segmentation Refinement

The partitions produced by the inference can contain layers which are spatially non-contiguous. To produce the final image segmentation we run connected components on the inference output. This splits up layers into spatially contiguous segments. All segments containing four or less super-pixels are merged with a larger spatial neighbor. If multiple larger neighbors exist, then the one which maximizes the appearance likelihood is chosen.

4 Learning from Human Segmentations

In the previous sections we have described the spatially dependent Pitman Yor process mixture model and made a case for how it captures important qualitative features of human segmentations. In this section, we provide methods for quantitatively calibrating the models to the appropriate human

segmentation biases. Specifically, we tune the model hyper-parameters to the human segmentations from the 200 training images of the Berkeley Segmentation Dataset (BSDS) [11]. We show that in spite of the inherent uncertainty in the segmentations of an image, we are able to learn important low level grouping cues.

It is worth noting that although, supervised learning is prevalent for training Markov Random Field (MRF) models for segmenting predefined predefined object categories [20], the parametrization and statistical properties of our layered Gaussian Process model are significantly different from that of discrete MRFs. Furthermore, image segmentation is a less constrained problem than the problem of segmenting out predefined object categories. As a result, the mapping between model parameters and human annotations is more subtle and trickier to infer. Learning nevertheless is both possible and effective for our proposed model, as outlined below.

4.1 Segment Size Distributions

For each image j in a given training database, let T_j denote the number of segmented regions, and $1 \geq a_{j1} \geq a_{j2} \geq \cdots \geq a_{jT_j} > 0$ their proportions of the image area. To compare these counts to $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha_a, \alpha_b)$, we also sort the sampled frequencies, producing a *two-parameter Poisson–Dirichlet* (PD) distributed partition $\widetilde{\boldsymbol{\pi}} = (\widetilde{\pi}_1, \widetilde{\pi}_2, \widetilde{\pi}_3, \ldots)$ satisfying $\widetilde{\pi}_k > \widetilde{\pi}_{k+1}$ with probability one [13]. These ordered histograms then allow the likelihood of the data under any Pitman-Yor model to be computed, producing maximum likelihood (ML) model parameters $\hat{\alpha} = (\hat{\alpha}_a, \hat{\alpha}_b)$. For the BSDS, the estimated parameters equal $\hat{\alpha}_a = 0.6, \hat{\alpha}_b = 3$.

4.2 Pairwise Grouping Probabilities

We would like to accurately segment images containing novel objects and materials. While we cannot expect our training data to provide examples of all important region appearance patterns, it does provide other important cues. In particular, via human segmentations we can can learn to predict the probability that *pairs* of superpixels (or image patches) occupy the same segment.

For every pair of superpixels, we consider several potentially informative low-level features: (i) pairwise Euclidean distance between superpixel centers; (ii) intervening contours, quantified as the maximal response of the probability of boundary (Pb) detector [12] on the straight line linking superpixel centers; (iii) local feature differences, estimated via log empirical likelihood ratios of χ^2 distances between superpixel color and texture histograms [15]. To model non-linear relationships between these four raw features and superpixel groupings, each feature is represented via the activation of 20 radial basis functions. Concatenating these gives a feature vector ϕ_{ij} for every superpixel pair i, j.

Defining a vector of regression weights f of the same dimension as ϕ_{ij}, the predicted probability that a given superpixel pair lies in the same segment equals

$$p(z_i = z_j \mid \phi_{ij}, f) = \sigma(f^T \phi_{ij}), \quad \sigma(a) = \frac{1}{1 + e^{-a}}. \tag{29}$$

We train this logistic regression model via MAP estimation of f under a Gaussian prior. Both the variance of this prior, and an appropriate bandwidth for the radial basis functions, were determined via cross-validation. When probabilities are chosen to depend only on the distance between superpixels, the distribution constructed in subsequent sections defines a generative model of image features. When these probabilities also incorporate contour cues, the model becomes a conditionally specified distribution on image partitions, analogous to a conditional random field [10].

4.3 From Grouping to Correlations

Our layered image model employs a sequence of support functions sampled from multivariate Gaussian distributions. These Gaussians, whose dimension equals the number of superpixels, have unit variance and a potentially different correlation ρ_{ij} for each superpixel pair i, j. For each superpixel pair, the probability that they lie in the same segment is *independently* determined by the corresponding correlation coefficient. As detailed in the appendix, using low-dimensional numerical integrations we can determine the probability that both superpixels are assigned to layer 1, or to layer 2, and so on. Summing these over all k then produces the overall probability of assignment

to the same layer, whatever its index. This process induces a one-to-one mapping between pairwise correlations ρ_{ij}, and probabilities q_{ij} that the pair of superpixels lie in a common segment. Applying this mapping produces a model corresponding to any given probability estimates. Figure 5 visualizes the learnt mapping.

Prior samples. Figure 4 displays samples drawn from the spatially dependent Pitman Yor process prior. Depending on the features used to estimate pairwise super-pixel correlations, qualitatively different partitions are produced. As expected, conditionally specified image specific partitions result in segmentations closer to "true" human segmentations. Also, note the effect of dimensionality on the quality of sampled partitions. With dimensionality the quality of partitions improve, at the expense of computational efficiency during inference.

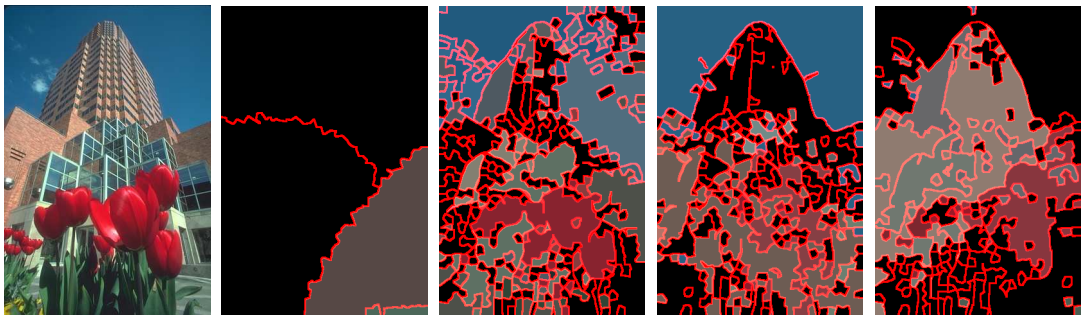

Figure 4: **Samples from various prior models.** Image partitions sampled from PY process assignments coupled by Thresholded GPs with different covariance functions. From left to right we present samples from (a)distance based GP covariance function (b) 100 dimensional projection of a GP covariance function learnt from low level features introduced in section 4.2 (These are used in all our experiments) (c) 500 dimensional projection (d) 1000 dimensional projection.

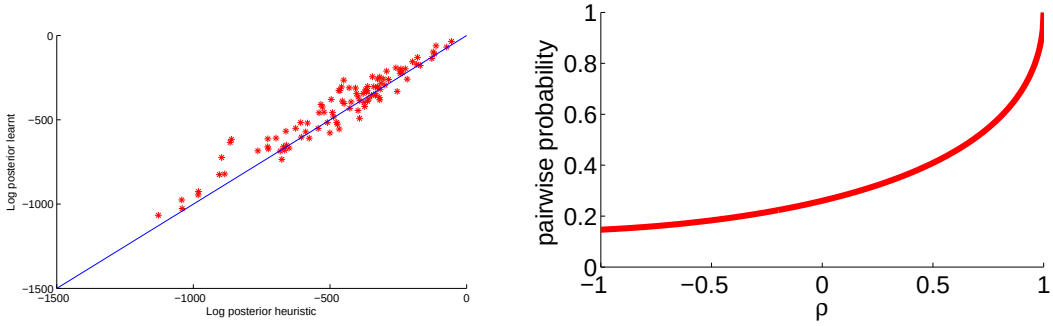

Figure 5: *Left.* Heuristic vs Learnt covariance functions. Each point in the plot represents one image from the BSDS test set. A majority of the points fall above the diagonal suggesting that higher posterior scores are achieved by the learnt covariance functions on a majority of the images. *Right.*Mapping between correlation coefficients and pairwise probabilities.

5 Experiments and Results

We benchmark our algorithm on the Berkeley Image Segmentation Dataset (BSDS300 [11]) and a set of images extracted from the LabelMe [16] dataset. BSDS300 contains 300 images, 200 training and 100 test images. The second dataset contains a subset of Olivia and Torralba's [19] eight natural categories dataset. We sampled 30 images at random from each of the eight categories to create a 240 image dataset.

The performance of the algorithms are quantified using the probabilistic Rand Index (*PRI*) and the segmentation covering (*SegCover*) metric introduced in [2]. To be consistent with [2] we report the

covering of a set of Ground truth segmentations by a machine produced segmentation. Furthermore, we present a number of segmentations discovered by our algorithm as a qualitative measurement of segmentation quality. In all experiments, our model *(PY-learnt)* used a 100 dimensional low rank representation and the corresponding inference utilized 100 discrete search iterations.

We start off by investigating the effect of the learnt covariance functions. On the 100 BSDS test images, we compare the log posterior marginal likelihoods of true human segmentations, under models using learnt and heuristic covariance functions. Figure 5 shows the corresponding scatter plot. As is evident, the learnt covariance functions assign higher posterior marginal likelihoods to human segmentations.

Next, we explore the effect of explicitly modeling the power law characteristic of human segment sizes. We compare against two spectral clustering based algorithms normalized cuts (Ncuts) [17] and multi-scale normalized cuts [4] algorithms. Ncuts biases all segments to have roughly equal size, while multi scale Ncuts somewhat relaxes this bias by incorporating a multi-resolution hierarchy. Both Ncuts and multi-scale Ncuts require the user to specify the number of segments to partition an image into, our algorithm being nonparametric does not. Here, we initialize both spectral algorithms with the average number of segments in the corresponding human segmentations of the image. The algorithms thus initialized, are denoted *ncuts-oracle* and *hncuts-oracle* in Table 1. Our algorithm outperforms both of these algorithms, in spite of the latter being tuned to the "true" number of segments per image. This gain in performance can be attributed to the fact that the region size statistics of our model matches the region size statistics of human segmentations.

Next, we quantify the performance PY-learnt and the proposed inference algorithm. We compare our model against the work of Sudderth and Jordan[18], who also employ thresholded dependent Gaussian processes to segment images. Remember that the work presented here improves upon [18] by using a better calibrated model and a more sophisticated inference algorithm. To quantify the effects of the proposed improvements, we compare our model and inference with three variants of previously proposed models

1. As a baseline we compare against the Bag of Features model (*BOF*).
2. The model and inference presented by Sudderth and Jordan (*Denoted by SJ in table 1*).
3. Sudderth and Jordan's model with our search based inference (*SJ+search*).

From table 1, we observe the efficacy of both our inference and the proposed model. *SJ+search* is significantly better than *SJ*, demonstrating the utility of the proposed inference algorithm. Combining the search based inference with the model proposed in this paper leads to a further performance jump which is close to state-of-the-art performance. However, we do note that the algorithm presented in [2] outperforms our algorithm on the BSDS300 test set.

Figure 9 illustrates yet another interesting property of our model, *layer ordering*. Remember that each image partition consists of a particular order of layers. Thus, in addition to recovering the most likely image partitions we also automatically recover the ordering of layers. Here, we illustrate some layer orders recovered by our algorithm. For the image on the left, the inferred ordering of the layers matches the true ordering of the objects in the scene. The images on the right illustrates a case when we infer an incorrect ordering. Since the model threshold's smooth GP functions it prefers explaining the generation of complex shapes through occlusion. As a result when an object in the foreground has a complicated shape, the model infers that it is more likely to have been generated as a result of occlusion and is moved back in the order.

Algorithms	PRI	SegCover
PY-learnt	0.77 ± 0.12	0.51 ± 0.02
SJ+search	0.71 ± 0.17	0.51 ± 0.17
ncuts-oracle	0.74 ± 0.14	0.34 ± 0.07
hncuts-oracle	0.75 ± 0.14	0.39 ± 0.08
SJ	0.49 ± 0.14	0.40 ± 0.01
BOF	0.46 ± 0.24	0.40 ± 0.20

Table 1: Quantitative performance of various algorithms on BSDS300

Figure 8 presents a set of diverse segmentations discovered by our algorithm. Although, our inference scheme searches for the MAP estimate, the search explores high probability regions of the distribution over partitions, hopping from partition to partition. In addition to the most likely partition, we also store other high probability partitions, leading to a richer description of the distribution over

Figure 6: **Segmentation Results.** Most likely segmentations for a number of BSDS300 test images

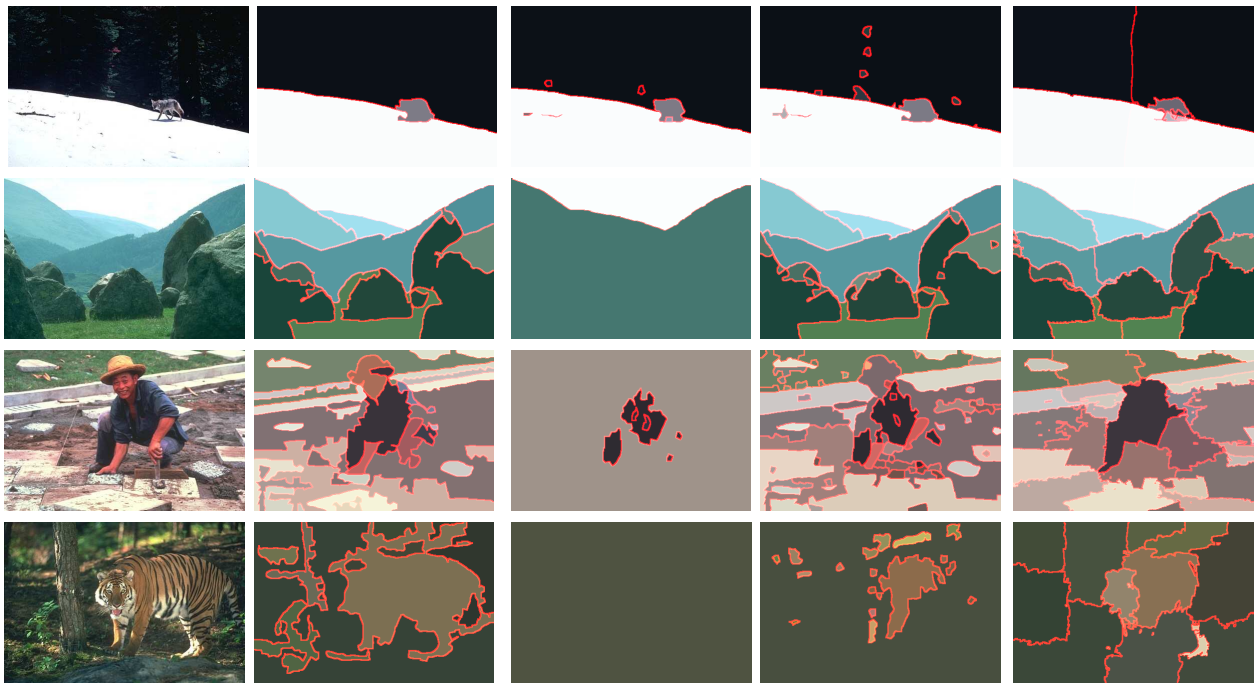

Figure 7: **Comparisons across models.** From Left to Right: Our model, BOF model,SJ+search, Multi scale Ncuts

partitions(segmentations). Figure 6 presents a random sub sampling of our results from BSDS300 for qualitative evaluation. **The complete set of segmentation results for the 240 LabelMe images can be found at `http://www.cs.brown.edu/~sghosh/results.html`**

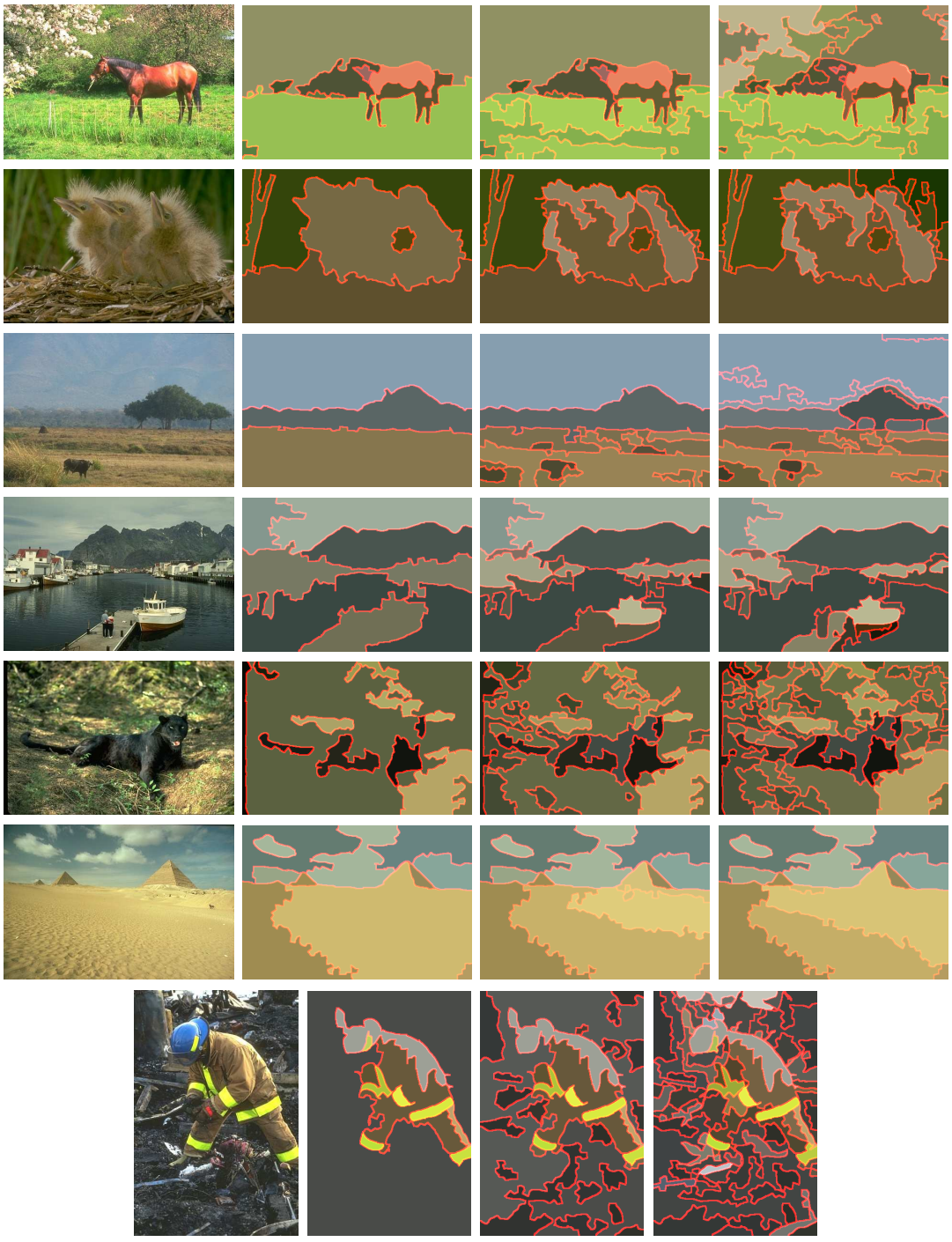

Figure 8: **Diverse Segmentations.** Diverse Segmentations discovered by our proposed algorithm. Each row depicts multiple segmentations for a given image. The segmentations are ordered in decreasing order of probability *(according to our model)* from left to right.

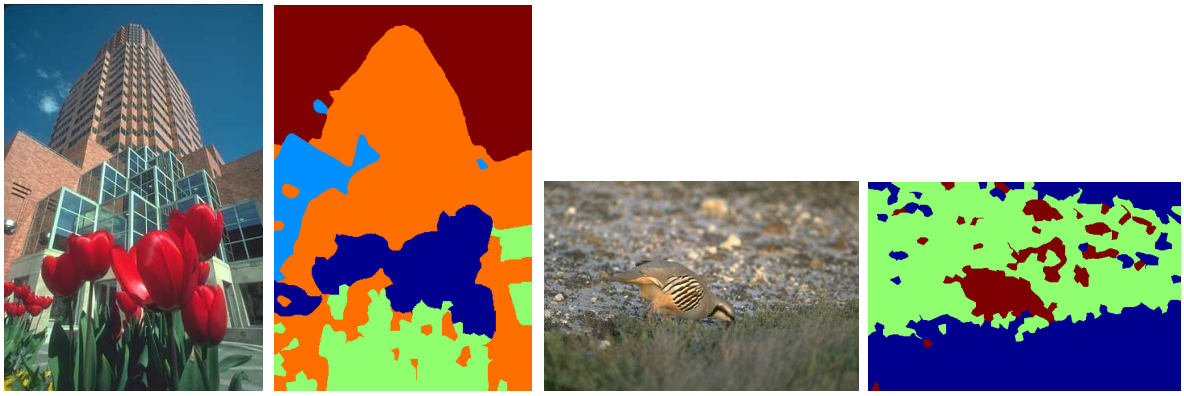

Figure 9: **Depth Ordering.** Blue segments are closest to the camera and red segments are farthest away. The two left images display a example where the algorithm infers the correct ordering. The two right images display a example where the wrong ordering of layers is inferred.

Algorithms	PRI	SegCover
gPb-ucm	0.68±0.17	0.54±0.17
PY-learnt	0.72±0.13	0.52±0.16

Table 2: Quantitative performance on LabelMe images

Finally, note that we tuned the hyper-parameters and not the parameters (there are none to tune) of our model to the Berkeley Training set. One would hope that tuning hyper-parameters would lead to better generalizability than algorithms such as the one presented in [2] (*gPb-ucm*) which tune model parameters via cross validation. To test this, we segmented the 240 LabelMe images using models tuned to the Berkeley dataset. Table 2 and figure 10 present the results of the comparison. While the segCover score achieved by [2] is higher than our algorithm, we significantly outperform them both qualitatively and in rand index scores.

6 Conclusion and Future Work

We have presented spatially dependent Pitman Yor process mixture models and developed an efficient, robust and accurate inference algorithm for these class of models. Further, we have shown the effectiveness of the presented model in partitioning complex natural scenes and its ability to model the inherent uncertainty found in human segmentations.

There are various natural extensions of this work. Our current models are limited to segment each image independently. In future work we plan on developing hierarchical versions of our models, which will collectively segment a group of images leveraging information from one image to help partition other images. Another aspect of the model which deserves attention is the appearance model. Our model currently uses naive color and texture histograms. As is standard practice, each bin of the histogram is considered independent. There is little justification for such independence assumptions other than computational ease. In fact, in natural images, different bins are often highly correlated. For instance, the color *white* and *blue* often occur together (blue sky with white clouds) suggesting that the corresponding bins should be positively correlated. We will address this issue in future work, by replacing the Dirichlet prior on the appearance histograms with a logistic Normal prior. Another exciting avenue of research is to further investigate the layer orderings recovered by the algorithm. Modeling shape should help the recovery of more accurate orderings. Finally, a long term goal of this work is to develop an image understanding system. Our current work only addresses the problem of "stuff" modeling and we are very interested in incorporating "thing" models and varying degrees of supervision to enable holistic natural scene interpretations.

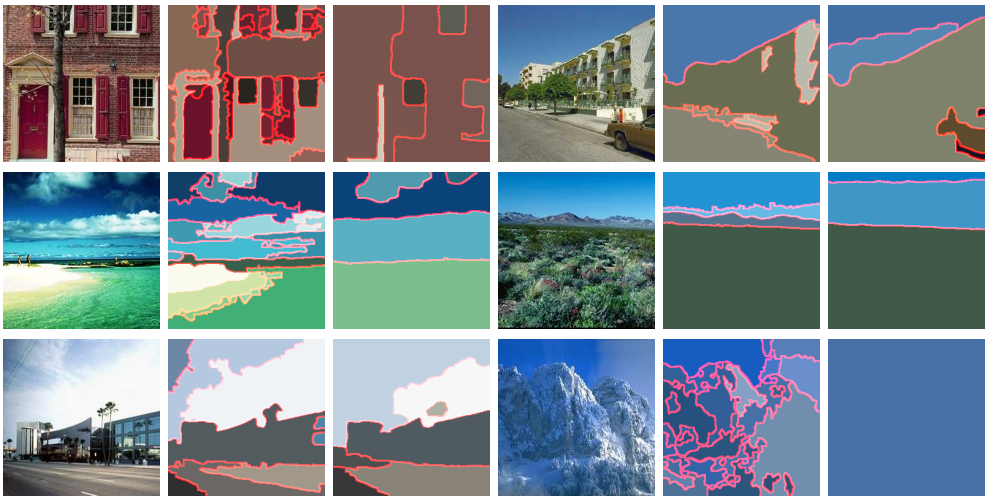

Figure 10: **Comparison on LabelME** From Left to Right: PY-learnt, gPb-ucm

References

[1] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In B. E. Rogowitz & T. N. Pappas, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4299 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 1–12, June 2001.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2294–2301, 2009.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[4] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 1124–1131, Washington, DC, USA, 2005. IEEE Computer Society.

[5] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *IJCV*, 72(2):195–215, 2007.

[6] J. A. Duan, M. Guindani, and A. E. Gelfand. Generalized spatial Dirichlet process models. *Biometrika*, 94(4):809–825, 2007.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.

[8] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. Technical report, Berkeley, CA, USA, 1996.

[9] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 30–43, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289. Morgan Kaufmann, 2001.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[12] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:530–549, May 2004.

[13] J. Pitman and M. Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997.

[14] X. Ren and J. Malik. A probabilistic multi-scale model for contour completion based on image statistics. In *ECCV*, volume 1, pages 312–327, 2002.

[15] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. 9th Int'l. Conf. Computer Vision*, volume 1, pages 10–17, 2003.

[16] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, 2008.

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 22(8), 2000.
[18] E. B. Sudderth and M. I. Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In *NIPS*, pages 1585–1592, 2008.
[19] A. Torralba and A. Oliva. Statistics of natural image categories. *Network*, 14:391–412, 2003.
[20] J. Verbeek and B. Triggs. Region classification with Markov field aspect models. In *CVPR*, 2007.
[21] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.
[22] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Tran. IP*, 3(5):625–638, Sept. 1994.
[23] M. Welling and K. Kurihara. Bayesian K-means as a "maximization-expectation" algorithm. In *SIAM Conf. Data Mining*, 2006.

7 Appendix

7.1 Likelihood Computation Details

We provide the solution to the color integral here for the sake of completeness (*To simplify notation we denote θ^c, x^c by just θ and x*).

For K segments and N super-pixels we have,

$$\int_\theta p(x|z,\theta)p(\theta|\rho^c)d\theta = \prod_{k=1}^{K}\int_{\theta_k} p(\theta_k|\rho^c)\prod_{n=1}^{N} p(x_n|z_n,\theta_k)^{\mathcal{I}(z_n=k)}d\theta_k \tag{30}$$

$$= \prod_{k=1}^{K}\int_{\theta_k} \Delta(\rho^c)\prod_{w=1}^{W_c}\theta_{kw}^{\rho_w^c-1}\prod_{n=1}^{N}\prod_{w=1}^{W_c}(\theta_{kw}^{x_{nw}})^{\mathcal{I}(z_n=k)}d\theta_k \tag{31}$$

$$= \prod_{k=1}^{K}\Delta(\rho^c)\int_{\theta_k}\prod_{w=1}^{W_c}\theta_{kw}^{\rho_w^c-1}\prod_{w=1}^{W_c}(\theta_{kw})^{\sum_n x_{nw}\times\mathcal{I}(z_n=k)}d\theta_k \tag{32}$$

$$= \prod_{k=1}^{K}\Delta(\rho^c)\int_{\theta_k}\prod_{w=1}^{W_c}(\theta_{kw})^{x_w^k+\rho_w-1}d\theta_k \tag{33}$$

$$= \prod_{k=1}^{K}\frac{\Delta(\rho^c)}{\Delta(\rho^c+x^k)} \tag{34}$$

In the above derivation $\Delta(\rho^c) = \frac{\Gamma(\sum_w \rho_w^c)}{\prod_w \Gamma(\rho_w^c)}$ and $x_w^k =$ number of times word w occurs with segment k.

7.2 Covariance Calibration Details

We are interested in estimating a mapping between the correlation (ρ) of a pair of Gaussian random variables (u_i and u_j), and the conditionally learned probability p_{ij} of the corresponding super-pixels i and j being assigned to the same layer. According to our generative model, two super-pixels i and j can be assigned to the same layer k iff both u_i and u_j are less than the threshold δ_k. Hence, the probability of two super-pixels being assigned to layer k is

$$p_-|\delta_k = \int_{-\infty}^{\delta_k}\int_{-\infty}^{\delta_k}\mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix}\middle|\middle|\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)du_i du_j \tag{35}$$

Furthermore, we can marginalize out the unknown thresholds δ_k

$$q_-^k(\alpha,\rho) = \int_{-\infty}^{\infty}\int_{-\infty}^{\delta_k}\int_{-\infty}^{\delta_k}\mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix}\middle|\middle|\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)p(\delta_k|\alpha)du_i du_j d\delta_k \tag{36}$$

Let us further define

$$q_+^k(\alpha,\rho) = \int_{-\infty}^{\infty}\int_{\delta_k}^{\infty}\int_{\delta_k}^{\infty} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \middle| \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right) p(\delta_k|\alpha) du_i du_j d\delta_k \tag{37}$$

which is the probability that both u_i and u_j are greater than the δ_k. Note that neither q_- nor q_+ can be computed in closed form and are both numerically approximated.

Now observe that two super-pixels i and j can be assigned to the same layer, if they are both assigned to the first layer or if neither is assigned to the first layer but both are assigned to the second layer or if neither is assigned to the first two layers but both are assigned to the third layer and so on. We can thus express p_{ij} as

$$\begin{aligned} p_{ij} &= q_-^1(\alpha,\rho) + q_-^2(\alpha,\rho)q_+^1(\alpha,\rho) + q_-^3(\alpha,\rho)q_+^1(\alpha,\rho)q_+^2(\alpha,\rho) + \ldots & (38)\\ &\approx \sum_{k=1}^{K} q_-^k(\alpha,\rho) \prod_{l=1}^{K-1} q_+^l(\alpha,\rho) & (39) \end{aligned}$$

where we have explicitly truncated our model to have K (some large number) layers. The above equation defines the sought relationship and allows us to map conditionally learnt p_{ij} to pairwise correlations of Gaussian random variables. The mapping is visualized in figure 5.