Nonparametric Density Estimation for Learning Noise Distributions in Mobile Robotics

David M. Rosen and John J. Leonard

***Abstract*—By admitting an explicit representation of the uncertainty associated with sensing and actuation in the real world, the probabilistic robotics paradigm has led to remarkable improvements in the robustness and performance of autonomous systems. However, this approach generally requires that the sensing and actuation noise acting upon an autonomous system be specified in the form of probability density functions, thus necessitating that these densities themselves be estimated. To that end, in this paper we present a general framework for directly estimating the probability density function of a noise distribution given only a set of observations sampled from that distribution. Our approach is based upon Dirichlet process mixture models (DPMMs), a class of infinite mixture models commonly used in Bayesian nonparametric statistics. These models are very expressive and enjoy good posterior consistency properties when used in density estimation, but the density estimates that they produce are often too computationally complex to be used in mobile robotics applications, where computational resources are limited and speed is crucial. We derive a straightforward yet principled approximation method for simplifying the densities learned by DPMMs in order to produce computationally tractable density estimates. When implemented using the infinite Gaussian mixture model (a specific class of DPMMs that is particularly well-suited for mobile robotics applications), our approach is capable of approximating *any* continuous distribution on $\mathbb{R}^n$ arbitrarily well in total variation distance, and produces C^∞, bounded, everywhere positive, and efficiently computable density estimates suitable for use in real-time inference algorithms on mobile robotic platforms.**

I. Introduction

By admitting an explicit representation of the uncertainty associated with sensing and actuation in a complex, noisy, and ever-changing world, the widespread adoption of the probabilistic robotics paradigm has led to remarkable improvements in the robustness and performance of autonomous systems. However, this approach generally requires [22] that the sensing and actuation noise acting upon an autonomous system be specified in the form of probability density functions, thus necessitating that these densities themselves be estimated.

In the context of mobile robotics, this density estimation problem is commonly addressed by assuming *a priori* that the true density is a member of a particular parametric family $p(\cdot|\theta)$ (e.g. Gaussian, gamma, exponential, etc.), and then attempting to estimate the parameter θ that best captures the observed noise characteristics. More advanced approaches (e.g. the Bayes [19] or Akaike [1] Information Criteria) allow joint model selection and parameter estimation from amongst a (possibly very large) finite set of candidate parametric families [5]. However, all of these techniques depend upon the practitioner's identification of a set of candidate parametric models containing at least one member that is able to capture the true noise characteristics well. It is far from clear that such a suitable *a priori* model selection will always be possible, especially in cases where the physical noise-generating process is complex or not well understood (cf. Fig. 1). This renders existing parametric density estimation approaches vulnerable to model misspecification, which can severely degrade the performance of autonomous systems that rely upon the resulting density estimates.

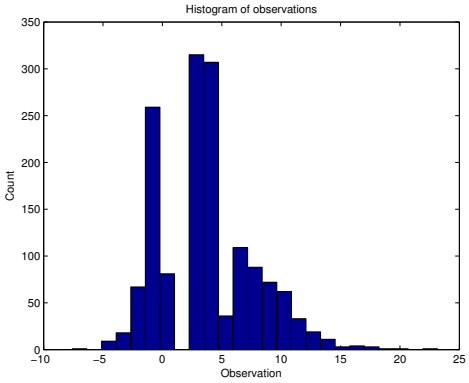


Fig. 1. A complicated noise distribution: This figure shows a histogram of 3000 observations sampled from a complicated noise distribution in simulation (cf. equations (37)–(40)). The *a priori* identification of a parametric family capable of accurately characterizing complex or poorly-understood noise distributions such as the one shown here is a challenging problem, which renders parametric density estimation approaches vulnerable to model misspecification. Nonparametric approaches overcome this difficulty by *inferring* an appropriate model directly from the data.

In this paper, we present a method for performing *non*parametric density estimation suitable for learning noise distributions in mobile robotics applications. Our approach is based upon *Dirichlet process mixture models* (DPMMs), a class of infinite mixture models commonly used in Bayesian nonparametric statistics [8]. These models are very expressive and enjoy good posterior consistency properties when used in density estimation [12], [25], but the density estimates that they produce are often too complex to admit real-time inference using the limited computational resources available on mobile robotic platforms. We derive a straightforward yet principled approximation method for simplifying the densities learned by DPMMs in order to produce computationally tractable density estimates. When implemented using the infinite Gaussian mixture model (a specific class of DPMMs that

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139, USA {dmrosen, jleonard}@mit.edu

is particularly well-suited for mobile robotics applications), our approach is capable of approximating *any* continuous distribution on $\mathbb{R}^n$ arbitrarily well in total variation distance, and produces C^∞, bounded, everywhere positive, and efficiently computable density estimates suitable for use in real-time inference algorithms on mobile robotic platforms.

II. The Dirichlet process

The *Dirichlet process* is a stochastic process whose samples are themselves random distributions. It was introduced by Ferguson in [8], who proposed its use in Bayesian nonparametric statistics as a prior over the set of all probability distributions on a given sample space. In this section, we review its formal definition and some of the properties that make it an attractive choice as a prior in Bayesian nonparametric applications.

A. Formal definition

Let Ω be a sample space, G_0 a probability measure on Ω, and $\alpha \in \mathbb{R}^+$ a positive scalar. A random probability distribution G on Ω is *distributed according to the Dirichlet process with base distribution G_0 and concentration parameter α*, denoted $G \sim \mathrm{DP}(\alpha, G_0)$, if

$$(G(A_1), \ldots, G(A_k)) \sim \mathrm{Dir}\left(\alpha G_0(A_1), \ldots, \alpha G_0(A_k)\right) \quad (1)$$

for every finite measurable partition $\{A_i\}_{i=1}^k$ of Ω. In other words, we require that the (random) probabilities assigned by G to elements of the partition $\{A_i\}_{i=1}^k$ be Dirichlet distributed.

B. Useful properties of the Dirichlet process

In this subsection we collect some useful properties of the Dirichlet process that make it an attractive choice as a prior in Bayesian nonparametric applications. Due to space constraints we present these results without proof; interested readers are encouraged to consult the excellent introductory articles [11], [20] for a more detailed exposition.

For the remainder of this section, we let Ω be a sample space, G_0 a probability measure on Ω, $\alpha \in \mathbb{R}^+$, $G \sim \mathrm{DP}(\alpha, G_0)$, and $\theta_1, \ldots, \theta_n \sim G$.

1) Expectation and variance: Letting $A \subseteq \Omega$ be a measurable subset and considering the partition $\{A, A^c\}$, we have from (1) that

$$G(A) \sim \mathrm{Beta}\left(\alpha G_0(A), \alpha(1 - G_0(A))\right) \quad (2)$$

and therefore that

$$\begin{aligned} E\left[G(A)\right] &= G_0(A), \\ \mathrm{Var}\left[G(A)\right] &= \frac{G_0(A) \cdot (1 - G_0(A))}{1 + \alpha}. \end{aligned} \quad (3)$$

Since (3) holds for all $A \subseteq \Omega$, this shows in particular that

$$E[G] = G_0. \quad (4)$$

In other words, the mean distribution (expectation) of the Dirichlet process is just the base distribution G_0. Thus, we can intuitively regard the Dirichlet process $\mathrm{DP}(\alpha, G_0)$ as a random, nonparametric "relaxation" of the base distribution G_0. The second equation in (3) also shows that the concentration parameter α controls the variance of the probability mass assigned by G to the set A; this justifies its nomenclature.

2) Posterior belief: The posterior belief for $G|\theta_1, \ldots, \theta_n$ is again Dirichlet-process distributed:

$$G|\theta_1, \ldots, \theta_n \sim \mathrm{DP}\left(\alpha + n,\ \frac{\alpha}{\alpha + n} G_0 + \frac{n}{\alpha + n} \cdot \frac{1}{n} \sum_{i=1}^{n} \delta_{\theta_i}\right) \quad (5)$$

where δ_{θ_i} is the measure assigning unit mass to the point θ_i. In other words, the Dirichlet process provides a class of priors over probability distributions G that is closed under posterior updates given observations $\theta_1, \ldots, \theta_n$ sampled from G.

3) Posterior predictive distribution: Given the observations $\theta_1, \ldots, \theta_n$, we can obtain the predictive distribution $\theta_{n+1}|\theta_1, \ldots, \theta_n$ for a subsequent sample θ_{n+1} drawn from G by observing that $\theta_{n+1}|G, \theta_1, \ldots, \theta_n \sim G$ and then marginalizing over G using the posterior distribution for $G|\theta_1, \ldots, \theta_n$ obtained in (5). This gives the posterior predictive distribution:

$$\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{\alpha}{\alpha + n} G_0 + \frac{n}{\alpha + n} \cdot \frac{1}{n} \sum_{i=1}^{n} \delta_{\theta_i}. \quad (6)$$

In other words, the posterior base distribution in (5) is also the predictive distribution for θ_{n+1} given $\theta_1, \ldots, \theta_n$.

4) Sample clustering and the Chinese restaurant process: Equation (6) implies an important clustering property for samples drawn from a random distribution G with a Dirichlet process prior. By the chain rule of probability, we can realize a sequence of draws $\theta_1, \ldots, \theta_n$ from G by sequentially sampling each θ_i according to

$$\theta_i \sim \theta_i|\theta_1, \ldots, \theta_{i-1}, \quad 1 \leq i \leq n, \quad (7)$$

where the right-hand side of equation (7) is the predictive distribution given in (6). Now, observe that the presence of the distributions δ_{θ_j} centered on the previous draws θ_j in the predictive distribution imply that $p(\theta_i = \theta_j|\theta_1, \ldots, \theta_{i-1}) > 0$ for all $1 \leq j < i \leq n$; i.e., *that previously sampled values will repeat with positive probability.*

More precisely, consider drawing the sample θ_i from the conditional distribution $\theta_i|\theta_1, \ldots, \theta_{i-1}$ at timestep i. Letting $\theta_1^*, \ldots, \theta_m^*$ denote the unique *values* within the set $\{\theta_1, \ldots, \theta_{i-1}\}$ of previously drawn samples and $n_1^*, \ldots, n_m^*$ denote their corresponding empirical counts, we have from (6) that θ_i repeats the value θ_j^* with probability

$$p(\theta_i = \theta_j^*|\theta_1, \ldots, \theta_{i-1}) = \frac{n_j^*}{\alpha + i - 1}, \quad (8)$$

and with probability $\alpha/(\alpha + i - 1)$ is a new value drawn from the base distribution G_0.

After sampling, the set $\theta_1, \ldots, \theta_n$ implicitly defines a partition of the set $\{1, \ldots, n\}$ into the clusters $\mathcal{C}_1, \ldots, \mathcal{C}_m$, where

$$\mathcal{C}_j = \{i \in \{1, \ldots, n\} \mid \theta_i = \theta_j^*\}, \quad 1 \leq j \leq m. \quad (9)$$

Observe that for the purposes of clustering, the actual *values* of the θ_j^* themselves do not matter; they simply function as *labels* for assigning elements of $\{1, \ldots, n\}$ to clusters. The corresponding algorithm for sampling such a partition is given as Algorithm 1.

Algorithm 1 The Chinese restaurant process

Initialize $\mathcal{C} \leftarrow \emptyset$.
for $i = 1 \dots n$ **do**
  Sample cluster label l according to the distribution:

$$p\left(l|\mathcal{C}, i\right) = \begin{cases} \frac{|\mathcal{C}_l|}{\alpha+i-1}, & \mathcal{C}_l \in \mathcal{C}, \\ \frac{\alpha}{\alpha+i-1}, & l = |\mathcal{C}|+1. \end{cases} \tag{10}$$

  if $\mathcal{C}_l \in \mathcal{C}$ **then**
    Add i to extant cluster $\mathcal{C}_l$: $\mathcal{C}_l \leftarrow \mathcal{C}_l \cup \{i\}$.
  else
    Initialize new cluster $\mathcal{C}_l$: $\mathcal{C}_l \leftarrow \{i\}$.
    Add $\mathcal{C}_l$ to cluster set: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{C}_l\}$.
  end if
end for
return $\mathcal{C}$

The sampling procedure implemented in Algorithm 1 is referred to as the *Chinese restaurant process* by analogy with the process of sequentially seating patrons at tables in a Chinese restaurant. This procedure provides an alternative view of the generative model underlying the Dirichlet process: To generate a set of samples $\theta_1, \dots, \theta_n \sim G$, where $G \sim \mathrm{DP}(\alpha, G_0)$, we may first sample a partition $\mathcal{C}$ for the set $\{1, \dots, n\}$ according to Algorithm 1, then draw $m = |\mathcal{C}|$ samples $\theta_1^*, \dots, \theta_m^* \sim G_0$, and finally assign $\theta_i = \theta_l^*$ for all $i \in \mathcal{C}_l$ and all $\mathcal{C}_l \in \mathcal{C}$.

The Chinese restaurant process also provides an alternative view of the concentration parameter α appearing in the Dirichlet process. Observe that during each iteration in Algorithm 1, i is assigned to a new cluster with probability $\alpha/(\alpha+i-1)$ *independent of all prior assignments* (cf. equation (10)). Therefore, the number of expected clusters $m = |\mathcal{C}|$ for a partition $\mathcal{C} \sim \mathrm{CRP}(\alpha, n)$ is

$$\begin{aligned} E[m|\alpha, n] &= \sum_{i=1}^{n} \frac{\alpha}{\alpha + i - 1} \\ &= \alpha \left(\psi(\alpha + n) - \psi(\alpha)\right) \end{aligned} \tag{11}$$

where ψ is the digamma function. The right-hand side of (11) can be approximated as

$$E[m|\alpha, n] \approx \alpha \ln\left(1 + \frac{n}{\alpha}\right), \quad n, \alpha \gg 0. \tag{12}$$

Thus we see that the concentration parameter α controls the asymptotic growth rate of the expected number of clusters as a function of the size n of the set to be partitioned.

As we will see, the clustering property of samples drawn from a distribution with a Dirichlet process prior (as encoded by the Chinese restaurant process) plays a crucial role in Bayesian nonparametric density estimation.

III. Dirichlet process mixture models

In this section we introduce Dirichlet process mixture models, a class of infinite mixture models commonly used in Bayesian nonparametric density estimation, and review algorithms for performing inference using these models.

A. Model definition

A *Dirichlet process mixture model* (DPMM) [7], [11] is a probabilistic generative model of the form

$$\begin{aligned} y_i|\theta_i &\sim F(\theta_i) \\ \theta_i|G &\sim G \\ G &\sim \mathrm{DP}(\alpha, G_0) \end{aligned} \tag{13}$$

where

- $F(\theta)$ is a parametric family of probability distributions supported on the observation space Ω with parameter space Θ and corresponding density function $f(\cdot;\theta)$,
- G_0 is a probability distribution supported on the parameter space Θ with density function g_0,
- $\alpha \in \mathbb{R}^+$ is the concentration parameter.

As shown in (13), DPMMs implicitly model observations $y_1, \dots, y_n \in \Omega$ as being drawn from a mixture of densities from the parametric class $F(\cdot)$. Since the parameter values θ_i specifying the mixture component $F(\theta_i)$ from which each observation y_i is sampled are themselves drawn from a distribution G with a Dirichlet process prior, some of the θ_i share common values with positive probability (as shown in Section II-B4), thereby leading to a partitioning of the observations y_i into clusters drawn from the same mixture component. Since the parameters $\theta = (\theta_1, \dots, \theta_n)$ are hidden variables within this model, then given a set $Y = (y_1, \dots, y_n)$ of observations, the posterior belief $p(\theta|Y)$ for the parameters θ also implicitly encodes the posterior belief over the assignment of the observations $y_1, \dots, y_n$ to clusters drawn from the same mixture component. In other words, DPMMs have the desirable property that the number of distinct mixture components appearing in the model is *not* fixed *a priori*, but rather is *inferred* from the data Y as part of learning the hidden parameters θ of the model. This effectively allows the inference procedure to *adapt* the model complexity (as measured by the number of inferred clusters/mixture components) to match the *observed* complexity of the data *as inference is performed*. This representational flexibility makes DPMMs an attractive choice for performing density estimation.

B. Nonparametric Bayesian density estimation using DPMMs

In this section we illustrate the use of DPMMs in Bayesian nonparametric density estimation.

We begin with a set of observations $Y = (y_1, \dots, y_n)$ that we assume are sampled from the generative model (13), and our goal is to learn the predictive distribution for a future sample y_{n+1}; i.e. we wish to learn the distribution for $y_{n+1}|Y$. Formally, we can compute this distribution as:

$$p(y_{n+1}|Y) = \int_{\Theta^n} p(y_{n+1}|\theta) \cdot p(\theta|Y)\, d\theta \tag{14}$$

where $\theta = (\theta_1, \dots, \theta_n) \in \Theta^n$ is the vector of hidden parameters in the generative model (13). The conditional distribution $p(y_{n+1}|\theta)$ in (14) can likewise be formally computed as:

$$p(y_{n+1}|\theta) = \int_{\Theta} p(y_{n+1}|\theta_{n+1}) \cdot p(\theta_{n+1}|\theta)\, d\theta_{n+1}. \tag{15}$$

Now the sampling distribution $p(y_{n+1}|\theta_{n+1})$ in (15) is just the density $f(y_{n+1};\theta_{n+1})$, which is specified as part of the model (13). Similarly, the predictive distribution $p(\theta_{n+1}|\theta)$ is given by (6). Substitution of these densities into (15) produces:

$$p(y_{n+1}|\theta) = \frac{\alpha}{\alpha+n}\int_{\Theta} f(y_{n+1};\theta_{n+1})\cdot g_0(\theta_{n+1})\,d\theta_{n+1} + \frac{1}{\alpha+n}\sum_{i=1}^{n} f(y_{n+1};\theta_i) \tag{16}$$

where g_0 is the density for the base distribution G_0. This gives the first of the two densities appearing in the integral in (14).

Unfortunately, the second density $p(\theta|Y)$ is not analytically tractable (its computation implicitly involves inference over the space of partitions on the set of n elements, a large combinatorial space). Therefore, we will resort to approximate inference via Markov chain Monte Carlo [2]. Luckily, the predictive distribution (6) can be used to derive an efficient Gibbs sampling algorithm for MCMC over DPMMs [15].

Define

$$\theta_{-i} = (\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_n). \tag{17}$$

In order to implement a Gibbs sampler to estimate $p(\theta|Y)$, we must determine a closed form for the conditional distributions $p(\theta_i|\theta_{-i},Y)$. Observe that

$$\begin{aligned} p(\theta_i|\theta_{-i},Y) &= p(\theta_i|\theta_{-i},y_i) \\ &\propto p(y_i|\theta_i)\cdot p(\theta_i|\theta_{-i}). \end{aligned} \tag{18}$$

Since the θ_i are assumed to be sampled i.i.d. from G, the *order* in which they are sampled doesn't actually matter; consequently, equation (6) actually gives the distribution $\theta_i|\theta_{-i}$ in a closed form. Substitution of (6) into (18) shows that

$$p(\theta_i|\theta_{-i},y_i) = \frac{\alpha}{\eta} f(y_i;\theta_i)\cdot g_0(\theta_i) + \frac{1}{\eta}\sum_{j\neq i} f(y_i;\theta_j)\delta_{\theta_j} \tag{19}$$

where η is the normalization constant

$$\eta = \alpha\int_{\Theta} f(y_i;\theta_i)\cdot g_0(\theta_i)\,d\theta_i + \sum_{j\neq i} f(y_i;\theta_j). \tag{20}$$

Equations (19) and (20) enable the implementation of a Gibbs sampler for estimating $p(\theta|Y)$, provided that it is possible to evaluate the integral in (20) and to draw samples directly from the distribution described by the density in (19). This will be the case whenever G_0 belongs to a family of conjugate priors for the parametric class $F(\cdot)$ from which we can sample easily.

We can further improve upon the computational efficiency of the Gibbs sampler by explicitly modeling the clustering of the parameters $\theta = (\theta_1,\ldots,\theta_n)$. We construct this more efficient sampler by breaking each Gibbs scan into two parts: first we update the clustering partition, and then we resample the parameter values assigned to each cluster. Let $\mathcal{C} = \{\mathcal{C}_1,\ldots,\mathcal{C}_m\}$ denote a partition of the hidden parameters $(\theta_1,\ldots,\theta_n)$ into clusters, let $c = (c_1,\ldots,c_n)$ denote the corresponding class labels (i.e. $c_i = l$ iff $\theta_i \in \mathcal{C}_l$) and let

Algorithm 2 Gibbs sampling algorithm to approximate $p(\theta|Y)$ via MCMC

1: Initialize $\theta \leftarrow (\theta_1,\ldots,\theta_n)$, $c \leftarrow (c_1,\ldots,c_n)$.
2: **for** $t = 1...N$ **do**
3: **for** $i = 1\ldots n$ **do** ▷ Resample class labels
4: Draw new class label for c_i from the conditional distribution $p(c_i = l|c_{-i},y_i,\theta^*)$ defined in (23). If the sampled label c_i corresponds to a new cluster, sample a new cluster parameter $\theta^*_{c_i}$ according to the posterior density

$$p\left(\theta^*_{c_i}|y_i\right) = \frac{f\left(y_i;\theta^*_{c_i}\right)\cdot g_0\left(\theta^*_{c_i}\right)}{\int_{\Theta} f\left(y_i;\varphi\right)\cdot g_0\left(\varphi\right)\,d\varphi} \tag{21}$$

and add it to θ^*. If this resampling step produces an empty cluster $\mathcal{C}_e$, remove it from $\mathcal{C}$ and its corresponding parameter θ^*_e from θ^*.
5: **end for**
6: **for** $\mathcal{C}_l \in \mathcal{C}$ **do** ▷ Recompute cluster parameters
7: Resample cluster parameter θ^*_l according to the posterior density:

$$p\left(\theta^*_l|c,Y\right) = \frac{\left(\prod_{c_i=l} f\left(y_i;\theta^*_l\right)\right)\cdot g_0\left(\theta^*_l\right)}{\int_{\Theta}\left(\prod_{c_i=l} f\left(y_i;\varphi\right)\right)\cdot g_0\left(\varphi\right)\,d\varphi}. \tag{22}$$

8: **end for**
9: Set $\theta^{(t)} \leftarrow \left(\theta^*_{c_1},\ldots,\theta^*_{c_n}\right)$ and $c^{(t)} \leftarrow (c_1,\ldots,c_n)$.
10: **end for**
11: **return** $\left\{\theta^{(1)},\ldots,\theta^{(N)}\right\}$, $\left\{c^{(1)},\ldots,c^{(N)}\right\}$

$\theta^* = \{\theta^*_1,\ldots,\theta^*_m\}$ denote the parameter values associated with each cluster (i.e. $\theta_i = \theta^*_{c_i}$ for all $1 \le i \le n$). By integrating (19), we obtain:

$$p(c_i = l|c_{-i},y_i,\theta^*) = \begin{cases} \frac{|\mathcal{C}_l - \{c_i\}|}{\eta} f(y_i;\theta^*_l), & 1 \le l \le m \\ \frac{\alpha}{\eta}\int_{\Theta} f(y_i;\varphi)\cdot g_0(\varphi)d\varphi, & l = m+1, \end{cases} \tag{23}$$

where η is the normalization constant defined in (20). Equation (23) provides the conditional sampling distribution used to update class labels during each Gibbs scan in the improved Gibbs sampler (Algorithm 2).

Finally, given the MCMC approximation $\{\theta^{(1)},\ldots,\theta^{(N)}\}$ for $p(\theta|Y)$ obtained via Algorithm 2, equations (14) and (16) imply that the corresponding density estimate for the predictive distribution $p(y|Y)$ (i.e., the density estimate $\hat{p}_y$ that we wish to learn) is:

$$\begin{aligned} \hat{p}_y(y) &= p(y|Y) \\ &= \frac{1}{N}\sum_{i=1}^{N} p\left(y|\theta^{(i)}\right) \\ &= \frac{\alpha}{\alpha+n}\int_{\Theta} f(y;\theta_{n+1})\cdot g_0(\theta_{n+1})\,d\theta_{n+1} \\ &\quad + \frac{1}{N\cdot(\alpha+n)}\sum_{i=1}^{N}\sum_{j=1}^{n} f\left(y;\theta^{(i)}_j\right). \end{aligned} \tag{24}$$

IV. Computationally tractable approximation of the Bayesian density estimate

While the nonparametric density estimation method of Section III-B is sufficiently flexible to admit learning a large class of distributions, this flexibility and robustness come at the expense of computational complexity. Even if we take advantage of clustering in the parameters $(\theta_1, \ldots, \theta_n)$, the number of unique kernels $f(\cdot;\cdot)$ appearing in the density estimate $\hat{p}_y$ in (24) still has a prior expected asymptotic growth rate of $N \cdot \alpha \ln(1 + n/\alpha)$ (cf. equation (12)). Even using relatively modest sample sizes N in the Gibbs sampler, this still presents an unacceptable computational burden if we wish to use the learned density in real-time robotics applications, where computational resources are limited and speed is crucial. To that end, in this section we derive a method for approximating the density estimate $\hat{p}_y$ learned via Algorithm 2 using a density $\hat{p}_a$ that is simple enough to admit real-time inference using the limited computational resources available on mobile robotic platforms.

The key insight here is that we need to use the (nonparametric) machinery of the DPMM only because the number of unique mixture components m needed in the inner summation in the final line of (24) to accurately represent the observed data Y is initially unknown. However, we observe that the Gibbs sampler in Algorithm 2 is (by design) able to propose models of varying complexity (as measured by the number of clusters m) in response to the observed complexity of the data Y as inference is performed. Assuming that the DPMM density estimate $\hat{p}_y$ is (at least weakly) consistent [12], [25] and that there is sufficient data Y available to well-characterize the unknown distribution p_y that we wish to learn, the posterior distribution $p(\theta|Y)$ will be sharply peaked around a small subset of the parameter space Θ^n that is consistent with the observations Y. Consequently, once the Markov chain implemented by the Gibbs sampler in Algorithm 2 has converged (i.e. after the burn-in phase), many of the samples $\theta^{(i)}$ will *individually* encode predictive distributions $p(\cdot|\theta^{(i)})$ (cf. equation (16)) that are close to the true underlying distribution p_y that we wish to learn.

Therefore, we approximate the density $\hat{p}_y$ in (24) by considering each of its component predictive distributions $p(y|\theta^{(i)})$ as *independent candidates for model selection*. Formally, the model that we should then choose as our approximation $\hat{p}_a$ is the one that minimizes the information divergence [6] with the true density p_y:

$$\begin{aligned} \hat{p}_a(\cdot) &= p(\cdot|\hat{\theta}), \\ \hat{\theta} &= \underset{\{\theta^{(1)},\ldots,\theta^{(N)}\}}{\operatorname{argmin}} D\left(p_y(\cdot)||p(\cdot|\theta^{(i)})\right), \end{aligned} \tag{25}$$

where (by definition) the information divergence is given by

$$\begin{aligned} D\left(p_y(\cdot)||p(\cdot|\theta^{(i)})\right) &= E_{p_y}\left[\log\left(\frac{p_y(y)}{p\left(y|\theta^{(i)}\right)}\right)\right] \\ &= E_{p_y}[\log p_y(y)] - E_{p_y}\left[\log p\left(y|\theta^{(i)}\right)\right]. \end{aligned} \tag{26}$$

Algorithm 3 Estimating an unknown density p_y given only a set of i.i.d. samples $(y_1, \ldots y_n, z_1, \ldots, z_k)$ drawn from it.

1: Learn the parameters $(\theta^{(1)}, \ldots, \theta^{(N)})$ for the Bayesian DPMM density estimate $\hat{p}_y$ in (24) using Gibbs sampling (Algorithm 2) with input data $Y = (y_1, \ldots, y_n)$.
2: Select the best approximation $\hat{p}_a$ for p_y from the candidate model set $\left\{p\left(y|\theta^{(i)}\right) \mid 1 \leq i \leq N\right\}$ using the selection criterion (29) evaluated on the holdout set $Z = (z_1, \ldots, z_k)$.
3: **return** $\hat{p}_y$, $\hat{p}_a$.

Unfortunately, we cannot compute the right-hand side of (26) directly, since it involves knowing the density function p_y, which is precisely what we are attempting to estimate. However, we observe that the first term on the right-hand side is a function of the unknown density p_y *only* (indeed, it is the negative entropy $H(p_y)$ of the unknown distribution p_y), and hence is *independent* of the model selection. Similarly, while we likewise cannot analytically compute the expectation in the second term (as it again depends upon knowing the density p_y), if we have access to a large number of samples drawn from p_y (as is generally the case with high-throughput robotic sensors), it *is* possible to compute an empirical estimate $\hat{L}(Z|\theta^{(i)})$ for this term according to:

$$\hat{L}\left(Z|\theta^{(i)}\right) = \frac{1}{k}\sum_{j=1}^{k} \log p\left(z_j|\theta^{(i)}\right) \tag{27}$$

using an *additional* set of samples $Z = (z_1, \ldots, z_k)$ drawn i.i.d. from p_y. Furthermore, by the strong law of large numbers, this estimator is strongly consistent:

$$\lim_{k\to\infty} \hat{L}\left(Z|\theta^{(i)}\right) \xrightarrow{\text{a.s.}} E_{p_y}\left[\log p\left(y|\theta^{(i)}\right)\right] \tag{28}$$

whenever the expectation on the right-hand side of (28) is finite (cf. Theorem 4 of [9]). Equations (25)–(27) then imply that the model selection criterion we should use is:

$$\begin{aligned} \hat{p}_a(\cdot) &= p(\cdot|\hat{\theta}), \\ \hat{\theta} &= \underset{\{\theta^{(1)},\ldots,\theta^{(N)}\}}{\operatorname{argmax}} \hat{L}\left(Z|\theta^{(i)}\right). \end{aligned} \tag{29}$$

In other words, we simply choose the model that maximizes the likelihood of the observations $Z = (z_1, \ldots, z_k)$. It is important to emphasize that Z is ***not*** part of the set Y that is used to learn the estimate $\hat{p}_y$ in Algorithm 2; it is an *additional* holdout set used for model selection. If we simply select the model $\theta^{(i)}$ that has the greatest likelihood on the original data Y, we run the risk of overfitting; performing model selection using the holdout set Z helps to ensure that the selected model generalizes well. Indeed, this approach is closely related to the use of cross-validation on holdout sets in other machine learning contexts in order to ensure good generalization of the learned models [3], [13].

Using the approximation/simplification scheme defined in (29), our end-to-end density estimation method is summarized as Algorithm 3.

V. The infinite Gaussian mixture model

The density estimation framework outlined in Algorithm 3 is quite general: it admits the use of any base distribution G_0 and any parametric family $F(\cdot)$ for which we can easily sample the posterior distributions defined by the densities in equations (21) and (22) and evalute the integral appearing in equation (24). In this section, we consider a specific instance of this general framework that is particularly well-suited for mobile robotics applications: the infinite Gaussian mixture model (IGMM) [7], [17].

A. Model definition

1) Kernels: We will adopt the multivariate Gaussian distributions $\mathcal{N}(\mu, \Sigma)$:

$$\begin{aligned} &p\colon \mathbb{R}^d \to \mathbb{R} \\ &p\left(y|\mu, \Sigma\right) = \frac{\exp\left(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\right)}{(2\pi)^{d/2}|\Sigma|^{1/2}} \end{aligned} \tag{30}$$

as our parametric family $F(\cdot)$.

2) Base distribution: Let $C^d = \{\Sigma \in \mathbb{R}^{d\times d} \mid \Sigma > 0\}$ denote the set of $d \times d$ positive definite matrices. We will let the base distribution G_0 be a member of the parameteric family of *normal-inverse Wishart distributions* $\mathrm{NIW}(\mu_0, \kappa_0, \Sigma_0, \nu_0)$; this a four-parameter family of distributions on $\mathbb{R}^d \times C^d$ with density

$$\begin{aligned} &p\colon \mathbb{R}^d \times C^d \to \mathbb{R} \\ &p(\mu, \Sigma|\mu_0, \kappa_0, \Sigma_0, \nu_0) = \frac{|\Sigma_0|^{\frac{\nu_0}{2}}\exp\left(-\frac{1}{2}\operatorname{tr}\left((\Sigma_0\Sigma^{-1})\right)\right)}{2^{\frac{d\nu_0}{2}}\Gamma_d\left(\frac{\nu_0}{2}\right)|\Sigma|^{\frac{\nu_0+d}{2}+1}} \\ &\qquad\times \frac{\exp\left(-\frac{\kappa_0}{2}(\mu-\mu_0)^T\Sigma^{-1}(\mu-\mu_0)\right)}{\kappa_0^{-1/2}(2\pi)^{d/2}}, \end{aligned} \tag{31}$$

where $\Gamma_d(\cdot)$ is the multivariate Gamma function.

3) Prior conjugacy and marginal likelihood: The normal-inverse Wishart distributions are a parametric family of conjugate priors for the class of multivariate Gaussians. Given $Y = (y_1, \ldots, y_n)$ with $y_1, \ldots, y_n \sim \mathcal{N}(\mu, \Sigma)$ i.i.d. and prior $(\mu, \Sigma) \sim \mathrm{NIW}(\mu_0, \kappa_0, \Sigma_0, \nu_0)$, the posterior distribution for $(\mu, \Sigma)|Y, \mu_0, \kappa_0, \Sigma_0, \nu_0$ is:

$$(\mu, \Sigma)|Y, \mu_0, \kappa_0, \Sigma_0, \nu_0 \sim \mathrm{NIW}\left(\mu_n, \kappa_n, \Sigma_n, \nu_n\right), \tag{32}$$

where the updated parameters in (32) are given by

$$\begin{aligned} \bar{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i, \\ S &= \sum_{i=1}^{n}(y_i-\bar{y})(y_i-\bar{y})^T, \\ \mu_n &= \frac{\kappa_0}{\kappa_0+n}\mu_0 + \frac{n}{\kappa_0+n}\bar{y}, \\ \kappa_n &= \kappa_0 + n, \\ \nu_n &= \nu_0 + n, \\ \Sigma_n &= \Sigma_0 + S + \frac{\kappa_0 n}{\kappa_0+n}(\bar{y}-\mu_0)(\bar{y}-\mu_0)^T \end{aligned} \tag{33}$$

(cf. [10, pg. 87]). Similarly, by marginalizing over the hidden parameters (μ, Σ), we obtain the observation marginal likelihood distribution $y|\mu_0, \kappa_0, \Sigma_0, \nu_0$:

$$y|\mu_0, \kappa_0, \Sigma_0, \nu_0 \sim \mathcal{T}\left(\nu_0 - d + 1, \mu_0, \frac{\kappa_0+1}{\kappa_0(\nu_0-d+1)}\Sigma_0\right) \tag{34}$$

(cf. [10, pg. 88]). Here $\mathcal{T}(\nu, \mu, \Sigma)$ denotes the parametric class of *multivariate Student-t distributions*, with corresponding density

$$\begin{aligned} &p\colon \mathbb{R}^d \to \mathbb{R} \\ &p(y|\nu, \mu, \Sigma) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)\left(1+\frac{1}{\nu}(y-\mu)^T\Sigma^{-1}(y-\mu)\right)^{-\frac{\nu+d}{2}}}{\Gamma\left(\frac{\nu}{2}\right)(\nu\pi)^{\frac{d}{2}}|\Sigma|^{1/2}}. \end{aligned} \tag{35}$$

B. Model justification

The infinite Gaussian mixture model enjoys several attractive properties that recommend its use for density estimation for mobile robotics applications. Perhaps its greatest advantage is its generality: Gaussian mixtures are capable of arbitrarily closely approximating *any* continuous distribution on $\mathbb{R}^n$ in total variation distance [14]. This model is also computationally quite expedient. The prior conjugacy between the normal-inverse Wishart and multivariate Gaussian distributions (cf. equations (32) and (33)), together with the existence of efficient algorithms for directly sampling from the normal-inverse Wishart distribution, means that computing and sampling from the posterior distributions in equations (21) and (22) can be done efficiently; this admits the computionally efficient implementation of the Gibbs sampler in Algorithm 2. Furthermore, the final density estimate $\hat{p}_a$ learned by applying Algorithm 3 using the IGMM is a finite linear combination of the kernel densities (30) and (35), which likewise has several attractive computational properties; in particular, $\hat{p}_a$ is C^∞, bounded, everywhere positive, and efficiently computable, which makes it ideal for use in real-time inference algorithms on mobile robotic platforms.

VI. Inference using the learned density estimates

Here we briefly discuss how the density estimates $\hat{p}_y$ and $\hat{p}_a$ learned by means of Algorithm 3 can themselves be used in inference algorithms in mobile robotics applications.

As shown in equations (24) and (29) (respectively), the learned models $\hat{p}_y$ and $\hat{p}_a$ are both finite mixtures of some number of the kernels $f(\cdot;\cdot)$ specified as part of the data defining the DPMM (each corresponding to one data cluster identified by the Gibbs sampler in Algorithm 2) and one additional component whose density is formally given by:

$$p(y|G_0) = \int_\Theta f(y;\theta)\cdot g_0(\theta)\,d\theta \tag{36}$$

(corresponding to the marginal distribution of an observation y arising from sampling a mixture component not represented in the training set data). Consequently, the learned models can be used in any inference algorithm able to operate on mixtures of densities of the form $f(\cdot;\cdot)$ and $p(y|G_0)$ as given in (36). The

selection of the parametric class $F(\cdot)$ and base distribution G_0 in the definition of a DPMM (13) may therefore impact the suitability of the resulting estimates $\hat{p}_y$ and $\hat{p}_a$ for use in combination with a *particular* inference algorithm that the practitioner wishes to apply.

For example, in the context of online maximum likelihood estimation over factor graphs (a situation that arises when considering e.g. the bundle adjustment [23] and robotic mapping [21] problems) recent work by the authors [18] admits the use of any parametric family $F(\cdot)$ and base distribution G_0 for which the densities $f(\cdot;\cdot)$ and $p(y|G_0)$ are C^1 and bounded. In the more specific case of the infinite Gaussian mixture model of Section V, the densities $f(\cdot;\cdot)$ are multivariate Gaussians, and therefore the method of Olson and Agarwal [16] could also be applied, provided that the individual components $f(\cdot;\cdot)$ appearing in the models have widely-separated modes and that one is willing to either neglect the Student-t density in (34) (corresponding to the marginal density in (36)) or approximate it using another Gaussian with a suitably large covariance matrix.

VII. Experimental results

In this section we illustrate the method of Algorithm 3 by applying it to a representative example problem using the infinite Gaussian mixture model of Section V. While here we consider learning a distribution on $\mathbb{R}$ (for ease of visualization), we point out that this method will work in Euclidean spaces of arbitrary dimension.

In order to provide a ground-truthed assessment of the proposed method, we consider data sampled from a known distribution in simulation. Specifically, we will attempt to learn the distribution determined by the probability density function:

$$\begin{aligned} p_y(y) = .3 \cdot p_e(-y; 1) + .4 \cdot p_b(y-3; 2, 2) \\ + .3 \cdot p_g(y-5; 2, 2), \end{aligned} \tag{37}$$

where $p_e(x;\lambda)$ is the exponential distribution with scale parameter λ:

$$p_e(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0, \end{cases} \tag{38}$$

$p_b(x;\alpha,\beta)$ is the beta distribution with shape parameters α and β:

$$p_b(x;\alpha,\beta) = \begin{cases} \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}, & 0 \leq x \leq 1, \\ 0, & \text{otherwise}, \end{cases} \tag{39}$$

and $p_g(x;k,\theta)$ is the gamma distribution with shape parameter k and scale parameter θ:

$$p_g(x;k,\theta) = \begin{cases} \frac{1}{\Gamma(k)\theta^k} x^{k-1} \exp\left(-\frac{x}{\theta}\right), & x > 0, \\ 0, & x \leq 0. \end{cases} \tag{40}$$

This is a challenging density to learn, as it is a mixture of three heterogeneous components, two of which are asymmetric, one-sided, and have much heavier tails than the Gaussian kernels used in the IGMM, and the third of which is compactly supported. Furthermore, since this distribution is obviously not a mixture of Gaussians itself, it serves as a useful demonstration of the universal approximation property of the IGMM described in Section V-B.

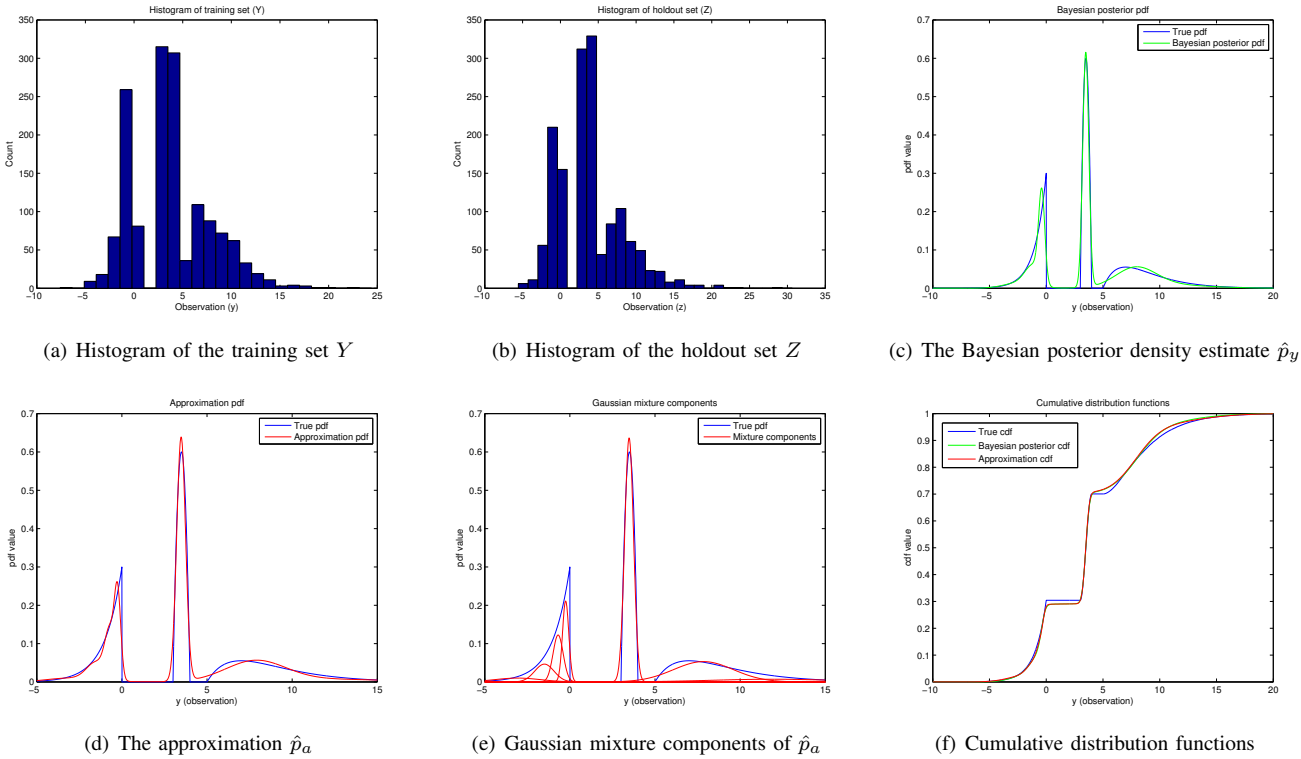
To run the experiment, we first drew 3000 samples from the distribution p_y defined in (37) (cf. Fig. 1). The first 1500 of these were used as the training data $Y = (y_1, \ldots, y_n)$ and the remaining 1500 as the holdout set $Z = (z_1, \ldots, z_k)$. To initialize the Gibbs sampler in Algorithm 2, we followed [17] by setting the base distribution parameters μ_0 and Σ_0 to be the sample mean and covariance of the training set Y, and let $\kappa_0 = 1$, $\nu_0 = 3$, and $\alpha = 1$ (since these parameters function as measures of prior confidence in the base distribution, we keep them small to indicate a relatively uninformative prior). We initialized the Markov chain by assigning every observation y_i to a single cluster $\mathcal{C}_1$, and drew an initial associated parameter θ_1^* directly from the base distribution. We then ran the sampler for 10000 iterations of burn-in, followed by another 10000 iterations to generate the samples $\{\theta^{(1)}, \ldots, \theta^{(N)}\}$. Finally, we performed model selection over this set of candidate models according to the selection criterion defined in (27) and (29). Results from this experiment are shown in Fig. 2.

To evaluate the performance of this method, we consider the information divergences $D(p_y||\hat{p}_y)$ and $D(p_y||\hat{p}_a)$. While the information divergence between two distributions does not, in general, have an analytically tractable solution, there are a variety of sample-based methods for estimating this quantity. In this case, we use the two-sample histogram-based method of [24] (specifically, Algorithm A with bias correction), which is a strongly consistent estimator of the information divergence under very general conditions (cf. Theorem 1 of [24]).

To perform the evaluation, we drew 100000 additional samples from each of p_y, $\hat{p}_y$, and $\hat{p}_a$, and used them to estimate $D(p_y||\hat{p}_y)$ and $D(p_y||\hat{p}_a)$. In this experiment, the Bayesian posterior predictive distribution $\hat{p}_y$ achieved an information divergence of $D(p_y||\hat{p}_y) = 0.082845$ and the approximation $\hat{p}_a$ achieved an information divergence of $D(p_y||\hat{p}_a) = 0.062366$, i.e., both of these models represent the underlying distribution p_y well, as illustrated by Fig. 2. However, the Bayesian model $\hat{p}_y$ uses 76185 Gaussian kernels, while the approximation $\hat{p}_a$ uses only 9. In other words, for this experiment, the model selection criterion (29) enabled us to reduce the computational complexity of the Bayesian posterior predictive distribution $\hat{p}_y$ by four orders of magnitude with minimal approximation loss.

VIII. Conclusion and future work

In this paper we present a general method for estimating the probability density function of a noise distribution given only a set of observations sampled from that distribution. Our approach is based upon Dirichlet process mixture models, but adapts these models to the context of mobile robotics through a simple approximation scheme that is able to reduce the computational complexity of the Bayesian posterior density estimates learned by these models by several orders of magnitude with minimal approximation loss. When implemented using the infinite Gaussian mixture model of Section V, our approach is capable of approximating *any* continuous distribution on $\mathbb{R}^n$



(a) Histogram of the training set Y

(b) Histogram of the holdout set Z

(c) The Bayesian posterior density estimate $\hat{p}_y$

(d) The approximation $\hat{p}_a$

(e) Gaussian mixture components of $\hat{p}_a$

(f) Cumulative distribution functions

Fig. 2. Learning the distribution p_y in equation (37). (a): Histogram of the training set Y (1500 samples). (b): Histogram of the holdout set Z (1500 samples). (c): The full Bayesian posterior predictive distribution $\hat{p}_y$ (76185 Gaussian kernels, $D(p_y||\hat{p}_y) = 0.082845$). (d): The approximation density $\hat{p}_a$ (9 Gaussian kernels, $D(p_y||\hat{p}_a) = 0.062366$). (e): The Gaussian mixture components of the approximation density $\hat{p}_a$. (f): Cumulative distribution functions for the true distribution p_y (blue) and the density estimates $\hat{p}_y$ (green) and $\hat{p}_a$ (red).

arbitrarily well in total variation distance, and produces C^∞, bounded, everywhere positive, and efficiently computable density estimates suitable for use in real-time inference algorithms on mobile robotic platforms.

It should come as no surprise that the development of more faithful models leads to better performance in inference. Our goal in pursuing this work is to provide a straightforward, flexible, and principled framework for learning these models that will in turn allow practitioners to take advantage of recent advances in inference algorithms for mobile robotics (e.g. [16], [18]) that enable the use of more general models than the customary single-Gaussian approximation. Our hope is that the more widespread adoption of expressive modeling techniques such as the one presented here (and their associated inference algorithms) will serve as a useful step towards attaining the representational and inferential robustness required for truly persistent robotic autonomy.

Finally, we mention several possible avenues for generalizing or improving the computational performance of the approach presented herein. In terms of generality, while the method of Algorithm 3 is suitable for learning marginal distributions of the form $p_y(y)$, we might also be interested in characterizing noise processes that we expect will exhibit some kind of state dependence; for example, a laser range scanner might exhibit different noise characteristics depending upon how far away the object being imaged is from the scanner. In these cases, it is more appropriate to learn conditional densities of the form $p_y(y|\varphi)$ for some set of configuration states $\varphi \in \Phi$. It would be interesting to attempt an extension of the current model to encompass learning these conditional distributions. Computationally, while the Gibbs sampler in Algorithm 2 enjoys many desirable properties (in particular, provable eventual convergence in distribution to the correct posterior $p(\theta|Y)$), running MCMC algorithms for long sequences N on large data sets Y can be computationally expensive, particularly since the approximation $\hat{p}_a$ only makes use of one of the resulting samples. For cases in which we would be satisfied with a point estimate of the model parameters (rather than desiring a full Bayesian posterior $p(\theta|Y)$ as in Algorithm 2), in which we are willing to consider only models $F(\cdot)$ that belong to an exponential family, and in which we are willing to tolerate the loss associated with the use of variational approximations (which can at least be measured *post hoc* by means of sample-based estimates of information divergence [24] in order to detect gross errors in the learned estimate), the use of variational inference techniques for DPMMs [4] can dramatically speed up the computation of the point estimate $\hat{p}_a$. We leave these considerations to future work.

ACKNOWLEDGMENTS

This work was partially supported by the Office of Naval Research under grants N00014-10-1-0936, N00014-11-1-0688, N00014-12-1-0020 and N00014-12-1-0093, which we gratefully acknowledge.

REFERENCES

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–723, Dec 1974.

[2] C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, 2003.

[3] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] D.M. Blei and M.I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

[5] K.P. Burnham and D.R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, 2nd edition, 2002.

[6] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

[7] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[8] T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[9] T.S. Ferguson. *A Course in Large Sample Theory*. Chapman & Hall/CRC, 1996.

[10] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.

[11] S. Ghosal. The Dirichlet process, related priors and posterior asymptotics. In *Bayesian Nonparametrics*, pages 36–83. Cambridge University Press, 2010.

[12] S. Ghosal, J.K. Ghosh, and R.V. Ramamoorthi. Posterior consistency of Dirichlet mixtures in density estimation. *Annals of Statistics*, 27(1):143–158, 1999.

[13] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA, 2009.

[14] G.J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[15] R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[16] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. In *Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.

[17] C.E. Rasmussen. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 12:554–560, 2000.

[18] D.M. Rosen, M. Kaess, and J.J. Leonard. Robust incremental online inference over sparse factor graphs: Beyond the Gaussian case. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013. (to appear).

[19] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[20] Y.W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*, pages 280–287. Springer, 2010.

[21] S. Thrun. Robotic Mapping: A Survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[22] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2008.

[23] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 153–177. Springer Berlin / Heidelberg, 2000.

[24] Q. Wang, S.R. Kulkarni, and S. Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9):3064–3074, 2005.

[25] Y. Wu and S. Ghosal. The L1-consistency of Dirichlet mixtures in multivariate Bayesian density estimation. *Journal of Multivariate Analysis*, 101(10):2411–2419, 2010.