Generative Supervised Classification Using Dirichlet Process Priors.

Manuel Davy, Jean-Yves Tourneret

Generative Supervised Classification Using Dirichlet Process Priors

Manuel Davy, *Member*, *IEEE*, and Jean-Yves Tourneret, *Senior Member*, *IEEE*

Abstract—Choosing the appropriate parameter prior distributions associated to a given Bayesian model is a challenging problem. Conjugate priors can be selected for simplicity motivations. However, conjugate priors can be too restrictive to accurately model the available prior information. This paper studies a new generative supervised classifier which assumes that the parameter prior distributions conditioned on each class are mixtures of Dirichlet processes. The motivations for using mixtures of Dirichlet processes is their known ability to model accurately a large class of probability distributions. A Monte Carlo method allowing one to sample according to the resulting class-conditional posterior distributions is then studied. The parameters appearing in the class-conditional densities can then be estimated using these generated samples (following Bayesian learning). The proposed supervised classifier is applied to the classification of altimetric waveforms backscattered from different surfaces (oceans, ices, forests, and deserts). This classification is a first step before developing tools allowing for the extraction of useful geophysical information from altimetric waveforms backscattered from nonoceanic surfaces.

Index Terms—Supervised classification, Bayesian inference, Gibbs sampler, Dirichlet processes, altimetric signals.

1 Introduction

Statistical classification has received considerable attention in the literature (see, for instance, [1], [2], [3], [4], [5], and references therein). Among all possible classification strategies, the Bayes decision rule is known to provide a minimal probability of error (or, more generally, a minimal expected value of an appropriate loss function). However, the Bayes classifier assumes knowledge of the class-conditional densities and the prior class probabilities. When the class-conditional densities are unknown, a classifier is generally built by using training patterns from the different classes following supervised learning. This paper focuses on supervised parametric classification. Each feature vector belonging to one of J identified classes is assumed to be the random outcome of some stochastic process tuned by an unknown parameter vector. In this setting, training the classifier consists of learning the parameter class-conditional posterior distribution (see Section 2) so as to compute Bayes factors [6]. We focus in this paper on generative classifiers that provide better performance than discriminative classifiers for small training set sizes [7].

Learning a parameter posterior distribution is generally a difficult problem. The parameter prior distributions have to be sufficiently general to model accurately the parameter class-conditional distributions without yielding untractable computational complexity. As a consequence, advanced learning strategies have to be developed to handle the possible multiple maxima of the objective function. These difficulties explain why fully Bayesian generative supervised classifiers have received less attention than other classifiers. A simple generative training strategy known as naive Bayes classification assumes that all features are independent inside a given class. This strategy has been used successfully in many practical applications, including text classification [8] and object detection [9]. Semi-naive Bayes classifiers decompose the input features into statistically independent subsets with statistical dependency within each subset [10]. This paper studies a new generative classifier based on Dirichlet process priors. These priors are known to enable the modeling of a large class of probability density functions (pdfs) (e.g., see [11], [12], [13], [14], [15], and references therein). The naive Bayes and semi-naive Bayes classifiers mentioned above are special cases of the proposed generative classifier based on Dirichlet process priors.

1.1 Contributions

A generative classifier based on hierarchical Bayesian models and Markov chain Monte Carlo (MCMC) methods has been recently studied in [16]. The Bayesian approach of [16] is adapted to cases where a precise generative data model can be defined. However, the classifier may be difficult to generalize to any classification problem and it can lead to high-computational complexity when conjugate priors cannot be used for the unknown parameters. The present paper studies an alternative supervised generative classifier involving Dirichlet process priors. Our approach assumes a nonparametric flexible parameter prior distribution with controlled complexity, known as the Dirichlet Process Mixture (DPM) [17]. Our main contribution is to derive DPM-based models for generative supervised classification as well as efficient MCMC algorithms, allowing one to estimate the unknown parameters of these models. It is interesting to mention here the work by Shahbaba and Neal [18], which is also related to supervised classification using DPMs. However, there are major

- *M. Davy is with VEKIA, 165 Avenue de Bretagne, 59000 Lille, France. E-mail: mdavy@vekia.fr.*
- *J.-Y. Tourneret is with the University of Toulouse, ENSEEIHT-IRIT-TéSA, 2 Rue Camichel, BP 7122, 31071 Toulouse Cedex 7, France. E-mail: jean-yves.tourneret@enseeiht.fr.*

differences between [18] and the proposed approach, as summarized below:

- Shahbaba and Neal have addressed classification in a discriminative context without using any generative model. Conversely, the proposed classification strategy studied here is generative allowing us to efficiently use the data and a parametric model associated to these data.
- Instead of modeling the parameter prior by a DPM, Shahbaba and Neal propose modeling the joint distribution of the training data and the class labels by a DPM.
- Shababa and Neal assume that the training data have a Gaussian distribution. This assumption is not necessary for the proposed algorithm. For instance, the radar data studied in the simulation section are distributed according to gamma distributions.

1.2 Paper Organization

A comprehensive glossary of the main notations is provided in Appendix A. Section 1 introduces the problem addressed in this paper. Section 2 presents the proposed parametric hierarchical Bayesian model used for supervised classification. Section 3 recalls some useful properties of Dirichlet processes. Section 4 studies original prior models for supervised classification. These models assume that the parameter priors for each class are mixtures of Dirichlet processes. Consequently, these models do not assume that the conditional class densities have a known parametric shape as is often the case in classification problems (see [2, p. 84] for details). Section 5 applies the proposed Bayesian methodology to the classification of synthetic and real data. Conclusions are reported in Section 6.

2 Bayesian Supervised Classification

This section recalls the principle of Bayesian supervised classification, using a parametric hierarchical model.

2.1 Problem Settings

Assume we want to classify (vectorial) data into J classes denoted as $C_1, \ldots, C_J$. We assume further that a training set is available for each class C_j, $j = 1, \ldots, J$, denoted as $\mathbf{X}_j = \{\mathbf{x}_{1,j}, \ldots, \mathbf{x}_{N_j,j}\}$. Inside each class, we assume the following generative model:

$$\mathbf{x}_{i,j} \;=\; M_j(\theta_{i,j}, \boldsymbol{\epsilon}_{i,j}), \tag{1}$$

where $M_j(\cdot)$ is a known function, the parameters $\theta_{i,j}$ live in a space $\Theta_j \subset \mathbb{R}^d$ (d is the dimension of the parameter space), $i = 1, \ldots, N_j$, and $\boldsymbol{\epsilon}_{i,j}$ is some stochastic noise. Note that the noise needs not be additive, e.g., the case of a multiplicative speckle noise will be considered in Section 5. In probabilistic terms,

$$\mathbf{x}_{i,j} \;\sim\; p_j(\mathbf{x}_{i,j}|\theta_{i,j}), \tag{2}$$

where $p_j(\mathbf{x}_{i,j}|\theta_{i,j})$ is the pdf of $\mathbf{x}_{i,j}$ (belonging to the class C_j) conditioned upon $\theta_{i,j}$.

The Bayesian algorithms, derived in this paper, estimate the posterior distribution of the parameter vector θ_j from the training set $\mathbf{X}_j$ for each class C_j, $j = 1, \ldots, J$. The parameter posterior distribution for class C_j will be denoted as $p(\theta_j|\mathbf{X}_j)$ and its estimation, detailed in the next section, will be referred to as Bayesian learning.

2.2 Parametric Hierarchical Approach

Within the parametric hierarchical Bayesian settings, learning $p(\theta_j|\mathbf{X}_j)$ requires the definition of a prior distribution over the parameter space Θ_j, denoted as $p_j(\theta_j|\boldsymbol{\phi}_j)$, with hyperparameter $\boldsymbol{\phi}_j$. The posterior distribution of θ_j can then be classically expressed as:[1]

$$p(\theta_j|\mathbf{X}_j) = \int p_j(\theta_j|\boldsymbol{\phi}_j) p(\boldsymbol{\phi}_j|\mathbf{X}_j) d\boldsymbol{\phi}_j, \tag{3}$$

where

$$\begin{aligned} p(\boldsymbol{\phi}_j|\mathbf{X}_j) &= \int \ldots \int p(\theta_{1,j}, \ldots, \theta_{N_j,j}, \boldsymbol{\phi}_j|\mathbf{X}_j) d\theta_{1,j} \ldots d\theta_{N_j,j} \\ &\propto \int \ldots \int p_j(\boldsymbol{\phi}_j) \prod_{i=1}^{N_j} \left[p_j(\mathbf{x}_{i,j}|\theta_{i,j}) p_j(\theta_{i,j}|\boldsymbol{\phi}_j)\right] \\ &\qquad d\,\theta_{1,j} \ldots d\,\theta_{N_j,j} \end{aligned} \tag{4}$$

by Bayes rule ($\propto$ means "proportional to"). As can be seen, a prior distribution over the hyperparameter, denoted as $p_j(\boldsymbol{\phi}_j)$, has to be defined to determine the posterior parameter distribution $p(\theta_j|\mathbf{X}_j)$ in (3). Once $p(\theta_j|\mathbf{X}_j)$ is known, a new observation vector $\mathbf{x}$ can be classified in view of the predictive densities

$$p(\mathbf{x}|\mathbf{X}_j) \;=\; \int p_j(\mathbf{x}|\theta_j) p(\theta_j|\mathbf{X}_j) d\theta_j, \tag{5}$$

for $j = 1, \ldots, J$. For instance, assuming equal class prior probabilities, the Bayesian maximum a posteriori (MAP) classifier assigns $\mathbf{x}$ to the class maximizing $p(\mathbf{x}|\mathbf{X}_j)$. The Bayesian algorithms derived in this paper estimate the posterior distributions $p(\theta_j|\mathbf{X}_j)$ from the training set $\mathbf{X}_j$.

2.3 Bayesian Computations

In this section, we drop the class subscript as the derivations are performed in each class in turn, independently of the other classes. Equivalently, we assume that the training samples in a class C_j do not provide any information regarding parameter θ_i for $i \neq j$ (see [2, p. 85] for a similar assumption). The classification of a given observation vector requires the evaluation of multidimensional integrals in (3) and (4). These integrals cannot be computed in closed-form in the general case. However, they can be evaluated by using Monte Carlo methods as proposed in [16]. The resulting algorithm is briefly recalled below (the reader is invited to consult [16] for more details).

Assume that a set of samples $\widetilde{\theta}^{(l)}$, $l = 1, \ldots, L$, distributed according to $p(\theta|\mathbf{X})$ is available. Note that $p(\theta|\mathbf{X})$ is defined as in (3), where the training set $\mathbf{X}_j$ has been replaced by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ for notation simplicity (the class subscript has been omitted). Then, the integral in (5) can be estimated as follows:

1. For the sake of notational clarity, we adopt a slight abuse of notation by denoting Lebesgue integrals over distributions as standard integrals. These integrals should be understood as sums whenever the parameter to be integrated is discrete.

$$p(\mathbf{x}|\mathbf{X}) \;=\; \int p(\mathbf{x}|\theta)p(\theta|\mathbf{X})d\,\theta \;\approx\; \frac{1}{L}\sum_{l=1}^{L} p\big(\mathbf{x}|\widetilde{\theta}^{(l)}\big). \quad (6)$$

A usual way of generating samples $\widetilde{\theta}^{(l)}$ distributed according to $p(\theta|\mathbf{X})$ is summarized in Algorithm 1.

Algorithm 1. Sampling from the class parameter posterior distribution
For $l = 1, \ldots, L$, do
1) sample $\widetilde{\boldsymbol{\phi}}^{(l)} \sim p(\boldsymbol{\phi}|\mathbf{X})$ given in (4)

2) sample $\widetilde{\theta}^{(l)} \sim g(\theta|\widetilde{\boldsymbol{\phi}}^{(l)})$

Algorithm 1 requires being able to sample from $p(\boldsymbol{\phi}|\mathbf{X})$. In order to do so, one generates samples distributed according to the joint posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{X})$, where $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$. The parameters θ_is are then marginalized out by only considering the component $\widetilde{\boldsymbol{\phi}}^{(l)}$ of the full set of sampled parameters and hyperparameters. In order to sample from the full posterior $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{X})$, one implements the following Gibbs sampler (where MH stands for "Metropolis-Hastings"):

Algorithm 2. Gibbs sampler for Bayesian Supervised classification
% Step 0: Initialization
- Sample $\widetilde{\boldsymbol{\phi}}^{(0)} \;\sim\; h(\boldsymbol{\phi})$

% step 1: Iterations. For $l = 1, \ldots, L$, do
- For $i = 1, \ldots, N$, sample $\widetilde{\theta}_i^{(l)} \sim p(\theta_i|\mathbf{x}_i, \widetilde{\boldsymbol{\phi}}^{(l-1)}) \propto p(\mathbf{x}_i|\theta_i)g(\theta_i|\widetilde{\boldsymbol{\phi}}^{(l-1)})$ using, e.g., an MH step

- Sample $\widetilde{\boldsymbol{\phi}}^{(l)} \sim p(\boldsymbol{\phi}|\widetilde{\boldsymbol{\theta}}^{(l)}) \propto p(\boldsymbol{\phi}) \;\prod_{i=1}^{N} g(\widetilde{\theta}_i^{(l)}|\boldsymbol{\phi})$ using, e.g., an MH step

2.4 Comments

Though straightforward to implement, Bayesian supervised classification requires the definition of several distributions. In practice, the distribution $p(\mathbf{x}|\theta)$ (referred to as likelihood) is imposed by empirical considerations. The prior $p(\theta|\boldsymbol{\phi})$ is chosen for its conjugacy as well as for physical considerations. However, its choice remains arbitrary in large parts. In the following, we propose a nonparametric parameter prior constructed from DPMs.

3 Dirichlet Processes and Dirichlet Process Mixtures

3.1 Dirichlet Processes

Dirichlet processes are characterized by a base distribution F_0 defined on an appropriate measurable space $(\Theta, \mathcal{B})$ and a positive real number α. More precisely, a Dirichlet process with base distribution F_0 and concentration parameter α, denoted as $\mathrm{F} \sim \mathcal{DP}(\alpha, \mathrm{F}_0)$, is the distribution of a random probability measure F over $(\Theta, \mathcal{B})$ such that, for Borel sets $A_1, \ldots, A_m$ forming a partition[2] of Θ,

$$[\mathrm{F}(A_1), \ldots, \mathrm{F}(A_m)] \;\sim\; \mathcal{D}\big(\alpha\mathrm{F}_0(A_1), \ldots, \alpha\mathrm{F}_0(A_m)\big), \quad (7)$$

where $\mathcal{D}$ is the standard Dirichlet distribution [17], [19]. Equivalently, a Dirichlet process F can be defined as a discrete distribution

$$\mathrm{F} \;=\; \sum_{k=1}^{\infty} \omega_k \delta_{U_k}, \quad (8)$$

where the probabilities ω_k are scalar nonnegative values that sum to one and the U_k are random variables living in the space of the base distribution. The random variables $(U_1, U_2, \ldots)$ are independent of the probabilities $(\omega_1, \omega_2, \ldots)$ and i.i.d. among themselves. They can be generated recursively as follows: $U_1 \;\sim \mathrm{F}_0$, $\omega_1 \sim\; \mathcal{B}(\omega_1; 1, \alpha)$, and, for $k = 2, 3, \ldots$:

- $U_k \;\sim \mathrm{F}_0(\cdot)$,
- $\beta_k \;\sim\; \mathcal{B}(\beta_k; 1, \alpha)$ and $\omega_k = \beta_k \prod_{k'=1}^{k-1}(1 - \beta_{k'})$,

where $\mathcal{B}(.; a, b)$ denotes the beta distribution with parameters a and b (defined in [2, p. 109]). This procedure is usually referred to as *stick breaking representation* [11], [12], [20] and explains the role of F_0 and α: F_0 determines the locations of the discrete components of F, while α determines the variance of the ω_ks. The reader is invited to consult [21] to understand the role of parameter α. A first analysis is conducted in [21] with a fixed value of α, i.e., $\alpha = 1$. The parameter α is then varied to understand how sensitive the results are to this parameter.

3.2 Polya Urn Representation

Consider a set of independent identically distributed (i.i.d) variables $\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N\}$ distributed according to $\mathrm{F}(\cdot)$, where $\mathrm{F} \sim \mathcal{DP}(\alpha, \mathrm{F}_0)$. Then, for any $i = 1, \ldots, N$,

$$\mathrm{p}(\boldsymbol{\phi}_i|\boldsymbol{\phi}_{-i}, \mathrm{F}_0, \alpha) \;=\; \frac{\alpha}{\alpha + N - 1}\mathrm{F}_0(\boldsymbol{\phi}_i) + \frac{1}{\alpha + N - 1}\sum_{i'=1}^{N}{}_{i' \neq i} \delta_{\boldsymbol{\phi}_{i'}}(\boldsymbol{\phi}_i), \quad (9)$$

where $\boldsymbol{\phi}_{-i} = \{\boldsymbol{\phi}_{i'}\}_{i'=1,\ldots,N,\; i' \neq i}$. This formula is known as the *Polya urn* representation, which also makes clear the role of F_0 and α [22]. The base distribution F_0 is used to sample the location of clusters. Large values of α cause many clusters to be created (the probability to sample from F_0 in (9) increases with α). Conversely, small values of α favor sampling from existing clusters and, overall, few clusters are generated. In theory, it is possible to have an infinite number of clusters. However, for a small value of α, only a few of them have a significant weight ω_k, and the probability that at least one $\boldsymbol{\phi}_i$ belongs to such a cluster becomes extremely small.

The Polya urn representation can also be written for the posterior distribution of F given $\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N\}$, as follows:

$$\mathrm{p}(\mathrm{F}|\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N, \mathrm{F}_0, \alpha) \;=\; \mathcal{DP}(\mathrm{F}_1, \alpha + N), \quad (10)$$

where

$$\mathrm{F}_1(\boldsymbol{\phi}) \;=\; \frac{\alpha}{\alpha + N}\mathrm{F}_0(\boldsymbol{\phi}) \;+\; \frac{1}{\alpha + N}\sum_{i=1}^{N} \delta_{\boldsymbol{\phi}_i}(\boldsymbol{\phi}). \quad (11)$$

3.3 Dirichlet Process Mixtures

A DPM is characterized by a DP F (with base distribution distribution F_0 and concentration parameter α) and a mixture distribution f. Precisely, a random variable θ is distributed according to a DPM whenever it is generated as follows:

2. A partition of a space Φ is a set of subsets $A_1, \ldots, A_m$ with arbitrary m such that $A_i \cap A_j = \emptyset$ for any $i \neq j$ and $A_1 \cup \ldots \cup A_m = \Phi$.

1. sample $\mathrm{F} \sim \mathcal{DP}(\alpha, \mathrm{F}_0)$,
2. sample $\boldsymbol{\phi} \sim \mathrm{F}(\cdot)$, and
3. sample $\theta \sim f(\theta|\boldsymbol{\phi})$.

By iterating Steps 2 and 3, one builds a family of random variables distributed according to a DPM with mixture distribution f, base distribution F_0, and parameter α. The role of f is to smooth the DP and to turn it into a continuous support distribution (whereas the DP F has discrete support). Each iteration i ($i = 1, \ldots, N$) of Step 2 yields a parameter $\boldsymbol{\phi}_i$ that determines the clusters θ_i. The Polya urn representation (9) shows that several $\boldsymbol{\phi}_i$s can share the same location. This can also be seen in the stick-breaking representation (8) emphasizing the locations U_ks and their probabilities ω_k.

Similarly to the finite mixture case studied in [23], it is possible to introduce a latent variable z_i to each θ_i ($i = 1, \ldots, N$). This consists of replacing Step 2 above by:

1. Sample $z \sim \mathrm{p}(z|\mathrm{F})$, where $\mathrm{p}(z = k|\mathrm{F}) = \omega_k$ and where ω_k comes from the stick-breaking representation of F.
2. Set $\boldsymbol{\phi} = U_z$, where U_z comes from the stick-breaking representation of F.

Finally, the following alternative representation of a DPM measure $\mathrm{G}(\theta)$ can be useful:

$$\mathrm{G}(\theta) \;=\; \int_{\Theta} F(\theta|\boldsymbol{\phi})\; d\mathrm{F}(\boldsymbol{\phi}) \text{ with } \mathrm{F} \;\sim\; \mathcal{DP}(\mathrm{F}_0, \alpha), \tag{12}$$

where $F(\theta|\boldsymbol{\phi})$ is the cumulative distribution function associated to the density $f(\theta|\boldsymbol{\phi})$.

4 Using DPMs to Model the Parameter Prior

This section introduces an original Bayesian supervised classification classifier based on DPM priors, referred to as DPM classifier. DPM classification consists of replacing the parameter prior distribution $\int_{\Phi} g(\theta|\boldsymbol{\phi})h(\boldsymbol{\phi})d\boldsymbol{\phi}$ by a DPM $\mathrm{G}(\theta)$. Motivations for using a DPM prior include the possibility of accurately describing a large variety of prior information with this mixture model. The DPM classification model for class C_i can be written:

$$\theta_i \sim \mathrm{G}, \tag{13}$$

$$\mathbf{x}_i \sim p(\mathbf{x}_i|\theta_i), \tag{14}$$

or, equivalently,

$$\mathrm{F} \sim \mathcal{DP}\big(\mathrm{F}_0, \alpha\big), \tag{15}$$

$$\boldsymbol{\phi}_i \sim \mathrm{F}, \tag{16}$$

$$\theta_i \sim f(\theta_i|\boldsymbol{\phi}_i), \tag{17}$$

$$\mathbf{x}_i \sim p(\mathbf{x}_i|\theta_i), \tag{18}$$

where $\mathrm{G}(\theta)$ is a DPM over Θ, with base distribution F_0 and mixture distribution f. Learning a DPM classifier consists of estimating the DPM parameter α, the parameters θ_i, and the hyperparameters $\boldsymbol{\phi}_i$ from the training set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

Example. In order to illustrate these elements, we propose the following toy example. Assume that θ is a random variable with Gaussian distribution, i.e., $f(\theta|\boldsymbol{\phi}) = \mathcal{N}(\theta; \mu, \sigma^2)$. The hyperparameter vector associated to this prior is $\boldsymbol{\phi} = [\mu, \sigma^2]$. Assume also that the base distribution F_0 is the product of a Gaussian distribution over μ and an inverse gamma distribution over σ^2. Then, it can be shown that, conditional on the parameters of F_0, θ is distributed according to a mixture of Gaussians

$$\mathrm{p}(\theta) \;=\; \sum_{k=1}^{\infty} \omega_k \mathcal{N}\big(\theta; \mu_k, \sigma_k^2\big), \tag{19}$$

where the weights are given by the stick-breaking rule and the parameters $\boldsymbol{\phi}_k = [\mu_k, \sigma_k^2]$ are distributed according to F_0, i.e., a Gaussian distribution for μ and an inverse gamma distribution for σ^2. This defines the prior distribution over θ. Note that the number of Gaussians defining the mixture is possibly infinite. In practice, only a finite number of Gaussians will have a significant weight, and the posterior distribution relies on these significant Gaussians. Therefore, the number of Gaussians is estimated in an implicit way. This example will be generalized to a multivariate parameter θ in Section 5.

4.1 Learning the Class Posterior Distribution

This section derives the posterior parameter distribution which follows from the model (15)-(18). By introducing the latent variables $\mathbf{z}$ and cluster locations $\mathbf{U}$, one has

$$\begin{aligned} p(\theta|\mathbf{X}) \;&=\; \int \sum_{\mathbf{z}} p(\theta, \mathbf{z}, \mathbf{U}|\mathbf{X}) d\mathbf{U} \\ &=\; \int \sum_{z,\mathbf{z}} f(\theta|U_z)\mathrm{p}(z|\mathbf{z})\mathrm{p}(\mathbf{U}, \mathbf{z}|\mathbf{X}) d\mathbf{U}, \end{aligned} \tag{20}$$

where $\mathbf{z} = (z_1, \ldots, z_N)$ is the indicator vector for $(\theta_1, \ldots, \theta_N)$, whereas z is the indicator variable for the parameter θ associated to the observation vector $\boldsymbol{x}$ to be classified. The distribution $\mathrm{p}(z|\mathbf{z})$ in (20) is defined as $\mathrm{p}(z = k|\mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{k,z_i}$, i.e., the number of latent variables z_i equal to k divided by the total number of latent variables (the number of samples in the training set). The parameter posterior distribution follows

$$\begin{aligned} \mathrm{p}(\mathbf{U}, \mathbf{z}|\mathbf{X}) &= \int_{\Theta^N} \mathrm{p}(\boldsymbol{\theta}, \mathbf{U}, \mathbf{z}|\mathbf{X}) d\boldsymbol{\theta} \\ &\propto \int_{\Theta^N} \mathcal{DP}(\mathrm{F}_0, \alpha) \prod_{i=1}^{N} p(\mathbf{x}_i|\theta_i) f(\theta_i|U_{z_i}) d\theta_i, \end{aligned} \tag{21}$$

where $\mathcal{DP}(\mathrm{F}_0, \alpha)$ is represented by its latent variables and cluster locations $(\mathbf{U}, \mathbf{z})$. Merging Algorithms 1 and 2 yields the following sampling strategy referred to as Algorithm 3 below.

Algorithm 3. Sampling from the class posterior distribution

% Step 0: Initialization

- For $i = 1, \ldots, N$, sample from the Polya urn $\widetilde{\boldsymbol{\phi}}^{(0)} \sim p(\boldsymbol{\phi}|\widetilde{\boldsymbol{\phi}}_1^{(0)}, \ldots, \widetilde{\boldsymbol{\phi}}_{i-1}^{(0)})$ and deduce $\widetilde{\mathbf{z}}^{(0)}$ and $\widetilde{\mathbf{U}}^{(0)}$
- For $i = 1, \ldots, N$, sample $\widetilde{\theta}_i^{(0)} \sim f(\theta|\widetilde{\boldsymbol{\phi}}_i^{(0)})$

% Step 1: Iterations For $l = 1, \ldots, L$, do

1.1- Sample $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{p}(\mathbf{z}|\widetilde{\mathbf{U}}^{(\mathbf{l-1})}, \widetilde{\boldsymbol{\theta}}^{(\mathbf{l-1})})$ using Algorithm 4 or Algorithm 5

1.2- Sample $\widetilde{\mathbf{U}}^{(l)} \sim p(\mathbf{U}|\widetilde{\mathbf{z}}^{(\mathbf{l})}, \widetilde{\boldsymbol{\theta}}^{(\mathbf{l-1})})$ using Algorithm 6

1.3- Sample $\widetilde{\boldsymbol{\theta}}^{(l)} \sim p(\boldsymbol{\theta}|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\mathbf{U}}^{(l)})$: for $i = 1, \ldots, N$, sample

$$\widetilde{\theta}_i^{(l)} \sim p(\theta_i|\mathbf{x}_i, \widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}}) \propto p(\mathbf{x}_i|\theta_i) f(\theta_i|\widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}})$$ using, e.g., a MH step

1.4- Sample $\widetilde{\xi}^{(l)} \sim \mathrm{p}(\xi|\widetilde{\mathbf{z}}^{(l)})$ such that $\mathrm{p}(\xi = k|\widetilde{\mathbf{z}}^{(l)}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{k,\widetilde{z}_i^{(l)}}$

1.5- Sample $\widetilde{\theta}^{(l)} \sim f(\theta|\widetilde{U}^{(l)}_{\widetilde{\xi}^{(l)}})$

1.6- (optional) Sample the DPM hyperparameters, i.e., α and the parameters defining F_0.

The algorithms used to sample the latent variables and the cluster locations are detailed in Appendix B. An important extension is Step 1.6, which consists of defining a prior for the hyperparameter α and the parameters of F_0. The parameter α can be sampled using the method described in [21] which assumes a gamma prior for α. The results of [21] show the robustness of the algorithm to α. It is important to note that the convergence of the algorithm is made easier by initializing α at a large value, and letting it decrease. This way, there are as many locations U_i as data $\mathbf{x_i}$ in the beginning of the algorithm. Then, during the iterations, Step 1.6 updates the value of α. At the very first iterations, this generally results in a decrease of its value (see Fig. 4), and therefore, the number of locations U_i also decreases. After some iterations, the value of α stabilizes. This procedure ensures good convergence because we make sure that all possible modes of the final distributions are included at the beginning of the algorithm and that the less significant are removed along the iterations. This somehow comes down to following the regularization path, starting from the less regularized solution, toward the most regularized solutions, and stabilizing around the appropriate regularization level. The parameters of F_0 can be sampled as shown in Section 5.

5 Application to the Classification of Synthetic and Real Data

The efficiency of the proposed DPM classifier has been tested on different synthetic and real data sets. This section summarizes some experiments that illustrate the interest of the proposed classification strategy.

5.1 Classification of Radar Altimeter Signals

This section studies a problem, recently introduced in [24], consisting of classifying radar altimeter data. The objective is to validate the proposed DPM-based classification method on real data. The use of altimetry measurements on ocean surface and continental ice has demonstrated its effectiveness. However, the quality of the onboard algorithms also allows the acquisition of several measurements over non-ocean surfaces (coastal areas and inland water). The corresponding measurements are being considered bad, although they probably contain useful geophysical information. Consequently, new processing algorithms efficient for all kinds of surfaces need to be developed. A first identified step before developing such processing algorithms is the classification of altimeter waveforms, according to the overflown surface. More precisely, this paper addresses the problem, recently introduced in [24], of classifying altimeter waveforms in the four classes "Ocean," "Desert," "Forest," and "Ice." The proposed DPM-based classification algorithm will show very promising results for this problem.

5.1.1 Statistical Model

Altimeter waveforms studied here represent the power of the backscattered altimeter echo. The observed time series are associated to the following statistical model:

$$\mathbf{x}(t) \;=\; M(t,\theta)\boldsymbol{\epsilon}(t), \quad t = 1, \ldots, T, \tag{22}$$

where $\mathbf{x} = [\mathbf{x}(1), \ldots, \mathbf{x}(T)]$ is the observed time series, $T = 104$ is the number of samples, and $\boldsymbol{\epsilon}(t)$ is an i.i.d. multiplicative noise referred to as *speckle* in the radar community (see, for instance, [25, p. 96], [26] for more details). Note that the generative model introduced in (1) reduces here to

$$\mathbf{x}_{i,j}(t) \;=\; M_j(t, \theta_{i,j}, \boldsymbol{\epsilon}_{i,j}) = M(t, \theta_{i,j})\boldsymbol{\epsilon}_{i,j}(t), \tag{23}$$

where the model $M_j = M$ is the same for all classes (the classes only differ by the distribution of the parameter vector θ) and where the dependence to the time instant t has been made explicit. The noise $\boldsymbol{\epsilon}(t)$ is known to be distributed according to a gamma distribution (see [27, p. 474] for the definition of the gamma distribution used in this paper), denoted as

$$\boldsymbol{\epsilon}(t) \;\sim\; \mathcal{G}a(\boldsymbol{\epsilon}(t); L, L), \tag{24}$$

where L is a known parameter (called *number of looks* in the radar community) associated to the altimeter. In this paper, we will use real data from the ENVISAT radar such that $L = 100$. The functional $M(t,\theta)$ in (22) results from the so-called Brown model introduced in [28] and described accurately in [29, Section 2.7.1.1]:[3]

$$\begin{aligned} M(t,\theta) = \; & P_n + \frac{a_\zeta \sigma_0}{2}\left[1 + \mathrm{erf}\left(\frac{t - \tau - c_\zeta \sigma_c^2}{\sqrt{2}\sigma_c}\right)\right] \\ & \exp\left[-c_\zeta\left(t - \tau - \frac{c_\zeta \sigma_c^2}{2}\right)\right], \end{aligned} \tag{25}$$

where

$$a_\zeta = \exp\left(\frac{-4\sin^2\zeta}{\xi_2}\right), \tag{26}$$

$$c_\zeta = \xi_1\left[\cos(2\zeta) - \frac{\sin^2(2\zeta)}{\xi_2}\right], \tag{27}$$

and where τ, σ_0, σ_c, and ζ are the unknown parameters, P_n is the known thermal noise level (it is estimated by using the first samples of the return waveform), and ξ_1, ξ_2 are fixed coefficients (depending on the antenna bandwidth, the mean satellite altitude, the velocity of light, and the earth radius). The $\mathrm{erf}(\cdot)$ function is the error function defined by $\mathrm{erf}(u) = 2/\sqrt{\pi}\int_0^u \exp(-v^2)dv$. The unknown parameters have the following physical interpretations (the interested reader is invited to consult [30] for more details):

3. The interested reader is invited to consult [29, Section 2.7.1.1] for more details. In particular, the full Hayne model includes additional terms which are negligible in the detection/classification problem, and are thus omitted here.

- $\tau > 0$ is the radar echo propagation time, related to the distance between the radar and the point observed on earth,
- $\sigma_0 > 0$ is the retrodiffusion coefficient depending on the ocean area,
- $\sigma_s > 0$ is related to the ocean significant wave height (SWH) by $\mathrm{SWH} = 2c_{\mathrm{light}}\sigma_s$ with $\sigma_s^2 = \sigma_c^2 - \sigma_p^2$ (c_{light} is the speed of light and σ_p^2 is a known coefficient),
- ζ is the radar antenna absolute off-nadir pointing angle.

The previous statistical model is fully characterized by the following parameter vector:

$$\theta = [\tau/\tau^{\mathrm{ref}}, \sigma_0/\sigma_0^{\mathrm{ref}}, \mathrm{SWH}/\mathrm{SWH}^{\mathrm{ref}}, \zeta/\zeta^{\mathrm{ref}}] \in \Theta \subset \mathbb{R}^4,$$

where each parameter has been scaled to ensure the components of θ are on a single scale (note that the dimension of the parameter space is $d = 4$ for this example). The pdf of the radar waveform satisfies the following relation:

$$\begin{aligned} p(\mathbf{x}|\theta) &= \prod_{t=1}^{T} \mathcal{G}a\left(\mathbf{x}(t); L, \frac{L}{M(t,\theta)}\right) \\ &\propto \exp\ -L\sum_{t=1}^{T}\left[\log M(t,\theta) + \frac{\mathbf{x}(t)}{M(t,\theta)}\right]\Bigg). \end{aligned} \tag{28}$$

5.1.2 Parameter and Hyperparameter Priors

The prior distribution for the unknown parameter vector θ is a DPM with mixture distribution $f(\theta|\boldsymbol{\phi})$, base distribution $\mathrm{F}_0(\boldsymbol{\phi})$, and DPM hyperparameter α. The mixture distribution is assumed to be a multivariate Gaussian distribution,

$$f(\theta|\boldsymbol{\phi}) = \mathcal{N}(\theta; \mu, \Sigma), \tag{29}$$

where $\boldsymbol{\phi} = \{\mu, \Sigma\}$ contains the hyperparameters (μ is the mean vector and Σ is the diagonal covariance matrix). For computational reasons, we assume a normal-inverse Wishart base distribution as in [31],

$$\mathrm{F}_0(\boldsymbol{\phi}) = \mathcal{NIW}(\mu, \Sigma; \mu_0, \kappa_0, \nu_0, \Lambda_0), \tag{30}$$

$$= \mathcal{N}(\mu; \mu_0, \Sigma/\kappa_0)\mathcal{IW}(\Sigma; \nu_0, \Lambda_0), \tag{31}$$

where μ_0, κ_0, ν_0, and Λ_0 are the base function hyperparameters (to which might be added the DPM hyperparameter α). The normal-inverse Wishart distribution is the conjugate prior for the mean and covariance matrix of a multivariate normal distribution. The reader is invited to consult [32, p. 272], [27, p. 80] for more details about this distribution. In particular, the inverse-Wishart distribution $\mathcal{IW}(\Sigma; \nu_0, \Lambda_0)$ has the following density:

$$\begin{aligned} \mathcal{IW}(\Sigma; \nu_0, \Lambda_0) \ &= \ \frac{2^{-\nu_0 d/2}\pi^{-d(d-1)/4}}{\prod_{i=1}^{d}\Gamma\left(\frac{\nu_0+1-i}{2}\right)} \frac{1}{|\Lambda_0|^{\nu_0/2}|\Sigma|^{(\nu_0+d+1)/2}} \\ &\exp\left[-\frac{1}{2}\mathrm{trace}\left(\Lambda_0^{-1}\Sigma^{-1}\right)\right]. \end{aligned} \tag{32}$$

5.1.3 DPM Posterior Distribution

Before proposing the algorithm to sample the parameters and hyperparameters according to their full posterior, we first note that the Gaussian distribution is conjugate for the normal-inverse Wishart distribution. Indeed,

$$\begin{aligned} &\left[\prod_{i=1}^{N} f(\theta_i|\boldsymbol{\phi})\right]\mathrm{F}_0(\boldsymbol{\phi}) \\ &\quad = \left[\prod_{i=1}^{N}\mathcal{N}(\theta_i; \mu, \Sigma)\right]\mathcal{NIW}(\mu, \Sigma; \mu_0, \kappa_0, \nu_0, \Lambda_0) \\ &\quad = C(\theta_{1:N})\,\mathcal{NIW}(\mu, \Sigma; \mu_N, \kappa_N, \nu_N, \Lambda_N), \end{aligned} \tag{33}$$

where

$$\overline{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i, \tag{34}$$

$$\mu_N = \frac{\kappa_0\mu_0 + N\overline{\theta}}{\kappa_0 + N}, \tag{35}$$

$$\kappa_N = \kappa_0 + N, \tag{36}$$

$$\nu_N = \nu_0 + N, \tag{37}$$

$$\begin{aligned} \Lambda_N^{-1} = \Lambda_0^{-1} &+ \sum_{i=1}^{N}(\theta_i - \overline{\theta})(\theta_i - \overline{\theta})^{\mathrm{T}} \\ &+ \frac{\kappa_0 N}{\kappa_0 + N}(\mu_0 - \overline{\theta})(\mu_0 - \overline{\theta})^{\mathrm{T}}, \end{aligned} \tag{38}$$

and the constant is (here, $d = 4$)

$$C(\theta_{1:N}) \ = \ \left[\frac{\kappa_0}{\kappa_N}\right]^{d/2} \pi^{-\frac{Nd}{2}} \frac{|\Lambda_N|^{\nu_N/2}}{|\Lambda_0|^{\nu_0/2}} \prod_{j=1}^{d}\frac{\Gamma\left(\frac{\nu_N+1-j}{2}\right)}{\Gamma\left(\frac{\nu_0+1-j}{2}\right)}. \tag{39}$$

This constant depends on $\theta_{1:N}$ through the parameter Λ_N and should be computed via its logarithm for numerical stability. Moreover, considering (33) for one θ, we have

$$\int f(\theta|\boldsymbol{\phi})\mathrm{F}_0(\boldsymbol{\phi})d\boldsymbol{\phi} \ = \ C(\theta) \tag{40}$$

since the normal-inverse Wishart integrates to one.

5.1.4 Sampling the Hyperparameters

Algorithm 3 contains an optional step for estimating the hyperparameter α of the DPM model and the hyperparameters defining the base distribution $\mathrm{F}_0(\boldsymbol{\phi})$. This optional step requires to define appropriate prior distributions for these unknown hyperparameters. This paper assumes that the prior for α is a gamma distribution as in [21]. The base distribution $\mathrm{F}_0(\boldsymbol{\phi})$ is a normal-inverse Wishart with parameters μ_0, κ_0, ν_0, and Λ_0. We assume here that μ_0 is known and we estimate κ_0, ν_0, and Λ_0. Therefore,

$$\begin{aligned} &\mathrm{p}(\kappa_0, \nu_0, \Lambda_0|\mathbf{U}, \mathbf{z}) \\ &\quad \propto \left[\prod_{z\in\mathcal{I}(z)}\mathrm{F}_0(U_z|\kappa_0, \nu_0, \Lambda_0)\right]\mathrm{p}(\kappa_0, \nu_0, \Lambda_0), \end{aligned} \tag{41}$$

where $\mathcal{I}(\mathbf{z})$ denotes the set of values taken by the variables $z_1, \ldots, z_l$ and $\mathrm{p}(\kappa_0, \nu_0)$ is the prior distribution of the hyperparameters. Assume the prior structure

$$\begin{aligned} \mathrm{p}(\kappa_0, \nu_0, \Lambda_0) \propto \mathcal{G}a(\kappa_0; \alpha_\kappa, \beta_\kappa)\mathcal{P}(\nu_0; \lambda_\nu)\mathcal{IW}(\Lambda_0; \nu_{00}, \Lambda_{00}) \\ \mathbb{I}_{\{d,d+1,\ldots\}}(\nu_0), \end{aligned} \tag{42}$$

where $\mathcal{P}(\nu_0;\lambda_\nu)$ is the Poisson distribution and $\mathbb{I}_{\{d,d+1,\ldots\}}(.)$ is the indicator variable for the set (d,∞). The hyperparameter posterior distribution can be decomposed into the terms

$$\mathrm{p}(\kappa_0|\mathbf{U},\mathbf{z}) = \mathcal{G}a(\kappa_0;\alpha'_\kappa,\beta'_\kappa), \tag{43}$$

$$\mathrm{p}(\nu_0|\Lambda_0,\mathbf{U},\mathbf{z}) \propto {\lambda'_\nu}^{\nu_0}\,\Gamma(\nu_0+1)^{-1} \prod_{i=1}^{d}\Gamma\left(\frac{\nu_0+1-i}{2}\right)^{-\#\mathcal{I}(\mathbf{z})}\mathbb{I}_{(d,\infty)}(\nu_0), \tag{44}$$

$$\mathrm{p}(\Lambda_0|\nu_0,\mathbf{U},\mathbf{z}) = \mathcal{IW}(\Lambda_0;\nu',\Lambda'), \tag{45}$$

where $\#\mathcal{I}(\mathbf{z})$ denotes the number of elements in $\mathcal{I}(\mathbf{z})$ and

$$\alpha'_\kappa = \alpha_\kappa + \frac{d\,\#\mathcal{I}(\mathbf{z})}{2}, \tag{46}$$

$$\beta'_\kappa = \beta_\kappa + \frac{1}{2}\sum_{z\in\mathcal{I}(\mathbf{z})}(\mu_z-\mu_0)^{\mathrm{T}}\Sigma_z^{-1}(\mu_z-\mu_0), \tag{47}$$

$$\lambda'_\nu = \lambda_\nu\, 2^{-\frac{d\#\mathcal{I}(\mathbf{z})}{2}}|\Lambda_0|^{-\frac{\#\mathcal{I}(\mathbf{z})}{2}}\prod_{z\in\mathcal{I}(\mathbf{z})}|\Sigma_z|^{-1/2}, \tag{48}$$

$$\nu' = \nu_0\#\mathcal{I}(\mathbf{z}) + \nu_{00}, \tag{49}$$

$$\Lambda'^{-1} = \Lambda_{00}^{-1} + \sum_{z\in\mathcal{I}(\mathbf{z})}\Sigma_z^{-1}. \tag{50}$$

These hyperparameters can be sampled conditionally on $\mathbf{U}$ and $\mathbf{z}$ in Step 1.6 of Algorithm 3 by direct sampling for κ_0 in (43) and Λ_0 in (45), and by an MH step for ν_0 with truncated Poisson proposal distribution.

5.2 Experiments

This section first presents simulation results obtained with synthetic data, in order to demonstrate the accurate learning ability of the proposed Bayesian model. Results obtained with real radar altimetry data are then analyzed.

5.2.1 Synthetic Data

The data studied here have been generated using the Brown model in (25). In order to demonstrate the good ability of our model to learn the posterior data distribution for a class, we consider one class with the following distribution for the model parameters (adjusted to match real data):

$$\tau \sim \mathcal{N}(\tau;32,4\cdot10^{-4}) \text{ and } \tau^{\mathrm{ref}} = 32, \tag{51}$$

$$\sigma_0 \sim \mathcal{N}(\sigma_0;12,2.25) \text{ and } \sigma_0^{\mathrm{ref}} = 12, \tag{52}$$

$$\mathrm{SWH} \sim 0.5\,\mathcal{G}a(\mathrm{SWH}-4;10,10) + 0.5\mathcal{U}\,(\mathrm{SWH};[1,2]) \text{ and } \mathrm{SWH}^{\mathrm{ref}} = 2, \tag{53}$$

$$\zeta \sim \mathcal{N}(\zeta;9.5\cdot10^{-4},6.3\cdot10^{-9}) \text{ and } \zeta^{\mathrm{ref}} = 0.001. \tag{54}$$

Fig. 1 shows 10 waveforms generated according to the Brown model in (25), with parameters sampled from the distributions (51)-(54) with multiplicative gamma-distributed white noise ($L = 100$). Step 1.3 of the above algorithm has been implemented using a one-at-a-time Gaussian random walk with variance $4\cdot10^{-4}$. The model hyperparameters have been selected as follows: $\mu_0 = [1\ 1\ 1\ 1]^{\mathrm{T}}$, $\alpha_\kappa = \beta_\kappa = 5$, $\lambda_\nu = d = 4$, $\Lambda_{00} = 0.1\mathbf{I}_d$, and $\nu_{00} = d+1$. These values are generic and changing them into a reasonable range does not much influence the results.

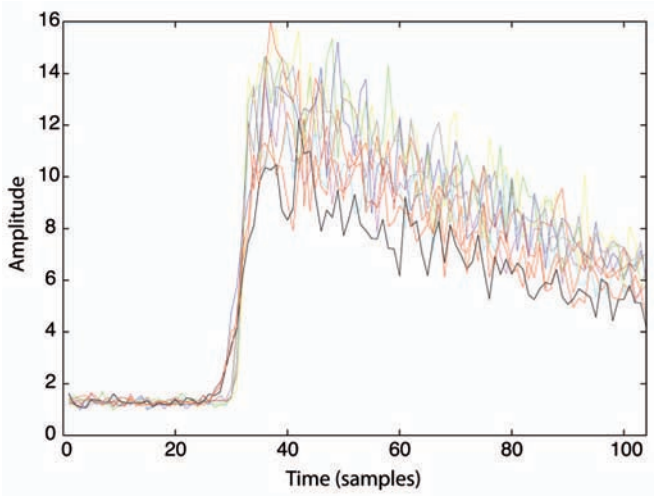

Fig. 1. Ten of the simulated radar altimetry time series (i.e., 10 data) used to train the classifier.

In order to check that the algorithm accurately learns the values of the different parameters for each training data set, the proposed algorithm has been run 100 times for $N = 100$ different training data sets. Each Monte Carlo run consists of $L = 5{,}000$ iterations (including $L_{\text{burn in}} = 3{,}000$ iterations). The minimum mean square error (MMSE) estimates of $\boldsymbol{\theta}_i^{\mathrm{gt}}$, denoted as $\widehat{\boldsymbol{\theta}}_i^{\mathrm{MMSE}}$, have been compared to their ground truth values via

$$\mathrm{Err}_{\boldsymbol{\theta}}(N) = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{\boldsymbol{\theta}}_i^{\mathrm{MMSE}}-\boldsymbol{\theta}_i^{\mathrm{gt}}\right)^{\mathrm{T}}\mathbf{S}^{-1}\left(\widehat{\boldsymbol{\theta}}_i^{\mathrm{MMSE}}-\boldsymbol{\theta}_i^{\mathrm{gt}}\right), \tag{55}$$

where $\boldsymbol{\theta}_i^{\mathrm{gt}}$ is the actual value of the parameter vector, $\mathbf{S}^{-1} = \mathrm{diag}(\tau^{\mathrm{ref}},\sigma_0^{\mathrm{ref}},\mathrm{SWH}^{\mathrm{ref}},\zeta^{\mathrm{ref}})$, and

$$\widehat{\boldsymbol{\theta}}_i^{\mathrm{MMSE}} = \frac{1}{L-L_{\text{burn in}}}\sum_{l=L_{\text{burn in}}+1}^{L}\widetilde{\boldsymbol{\theta}}_i^{(l)}, \tag{56}$$

where $\tilde{\theta}_1 = \tau/\tau^{\mathrm{ref}}$, $\tilde{\theta}_2 = \sigma_0/\sigma_0^{\mathrm{ref}}$, $\tilde{\theta}_3 = \mathrm{SWH}/\mathrm{SWH}^{\mathrm{ref}}$, and $\tilde{\theta}_4 = \zeta/\zeta^{\mathrm{ref}}$ are the normalized model parameters. The average of $\mathrm{Err}_{\boldsymbol{\theta}}(N)$ over the 100 Monte Carlo runs is 0.109 with standard deviation 0.063 (to be compared with the average norm of $\boldsymbol{\theta}_i^{\mathrm{gt}}$ over the 100 simulations, which is 5.43). Fig. 2 shows that the parameters of each training signal have been accurately learned through the simulations. Over the 100 simulations, only six failed to converge to the ground truth parameter values for each of the 100 parameters. In order to illustrate the algorithm convergence, Figs. 3 and 4 represent the simulated components of θ sampled in Step 1.5, the evolution of $\#\mathcal{I}(\mathbf{z})$, and the hyperparameters α, ν, and κ for one typical simulation with $N = 100$ data. The chain has clearly converged after about 3,000 burn-in iterations. Overall, the learning phase for this example takes a couple of minutes on a standard PC computer and Matlab/Octave code. Fig. 5 represents the histograms of the four components of the samples $\widetilde{\theta}^{(l)}$ for $l \in \{3{,}001,\ldots,5{,}000\}$. The

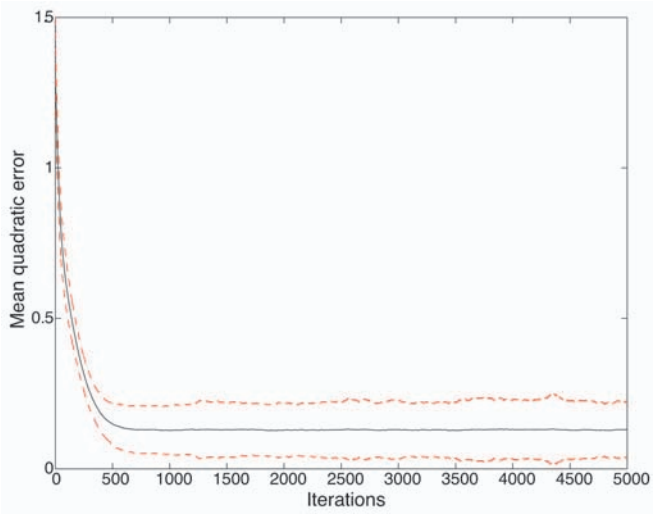

Fig. 2. Evolution of the mean quadratic error for the signal parameters $\theta^{(l)}$, $l = 1, \ldots, 100$ for each of the 5,000 iterations of the MCMC algorithm, averaged over 100 Monte Carlo runs. The upper and lower curves represent the 95 percent confidence interval (± 2 standard deviations).

estimated marginal posterior distributions are quite close to the true parameter distributions, which shows the ability of our model to learn multimodal, non-Gaussian distributions.

Fig. 6 shows the marginal posterior distributions of the different parameters for two sample sizes, $N = 10$ and $N = 1{,}000$, which have to be compared with Fig. 5, which has been obtained with $N = 100$. The higher N, the better the estimated parameter posteriors, as expected. When a very limited number of training data is available ($N = 10$), the learning ability is also correct. The regularization role of the DPM during learning can be clearly seen on the third component of θ: The model considers that there is only one mode for $N = 10$ (which is reasonable given the very limited number of training data) whereas the two modes are clearly recovered for $N = 100$ and $N = 1,000$. These simulations have clearly demonstrated the ability of the proposed algorithm to learn the parameter posterior distribution. The algorithm classification ability is then studied by using real radar altimetry data.

5.2.2 Real Data

As explained in [24], extraction of geophysical information from waveforms backscattered from nonoceanic surfaces is an identified goal of future altimetry missions. The classification of the received radar waveforms into the different classes

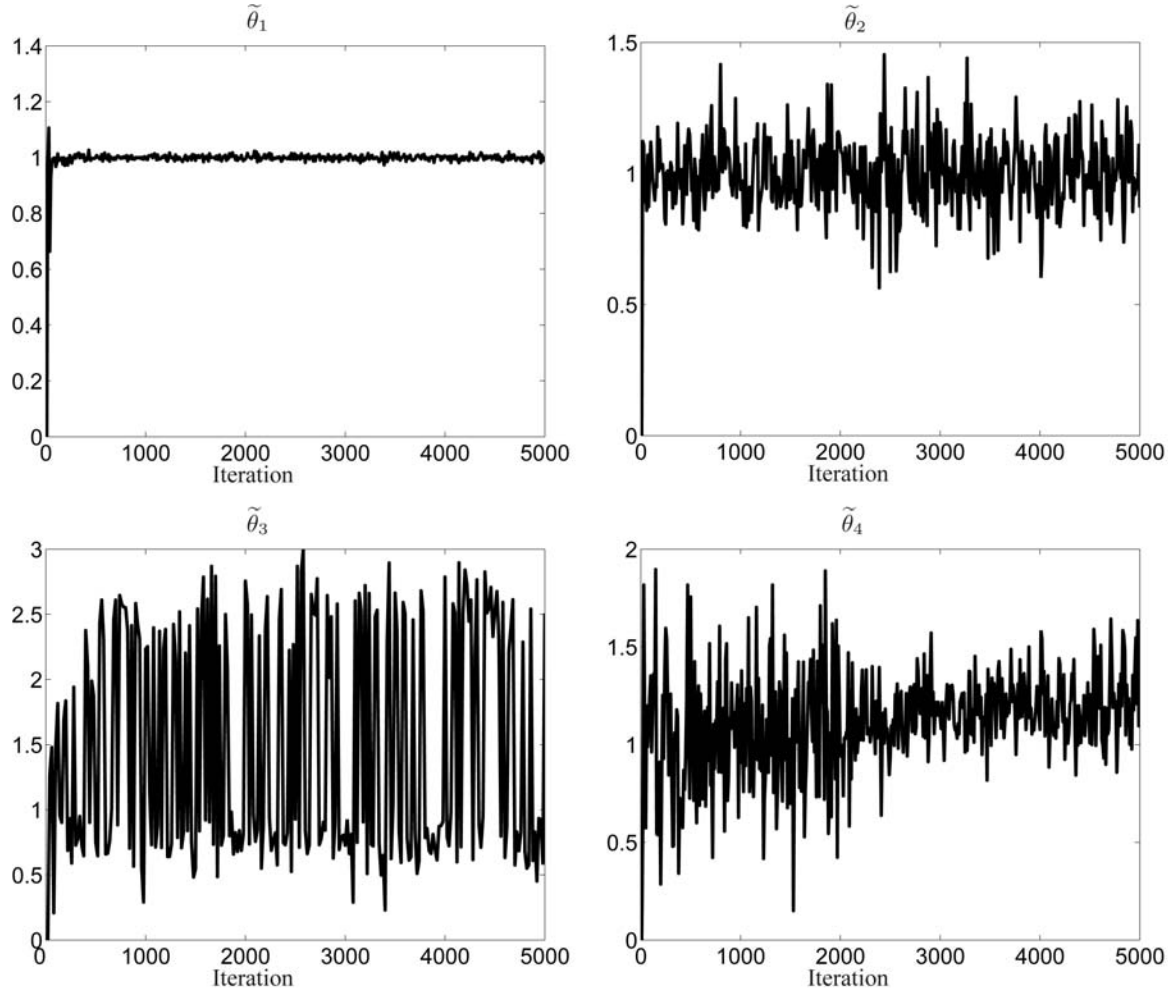

Fig. 3. Evolution of the parameter θ sampled at each iteration in Step 1.5 of Algorithm 3 from the class posterior distribution, for each of the four components. As can be seen, the MCMC algorithm mixes quite well.

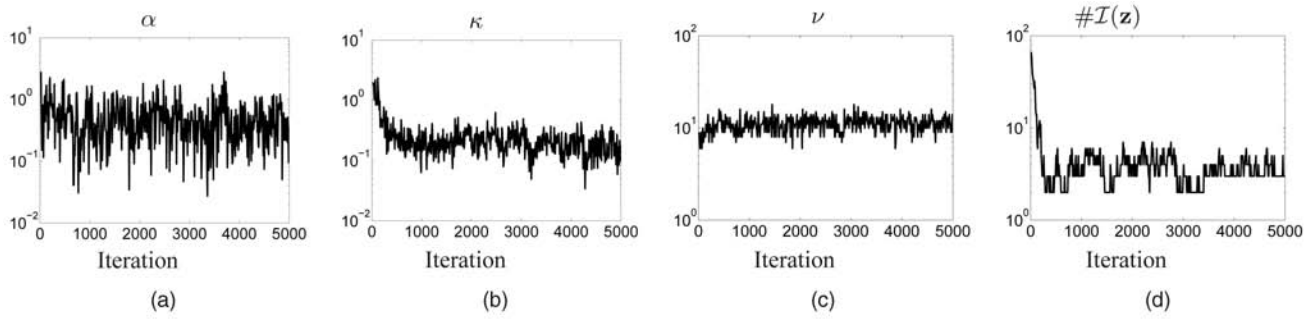

Fig. 4. Evolution of the hyperparameters α ((a) in log scale) , κ ((b) in log scale), ν ((c) in log scale) sampled at each iteration in Step 1.6 of Algorithm 3. (d) represents the evolution of the number of different values for z used at each iteration (in log scale).

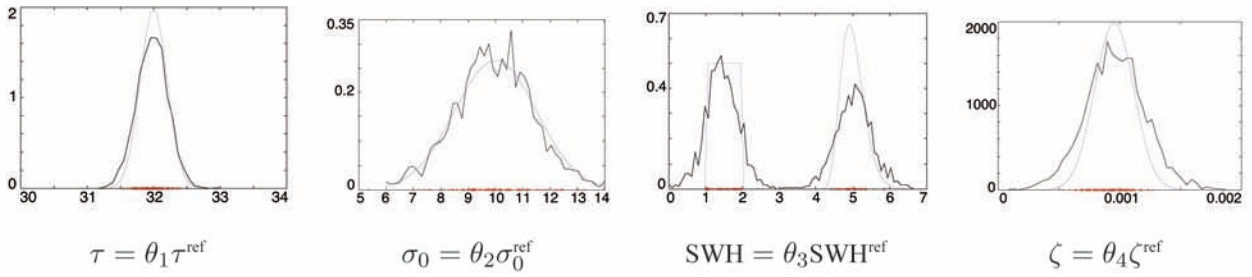

Fig. 5. Histograms of the four components of the parameter vector computed from the samples $\widetilde{\theta}^{(l)}$ for $l \in \{3{,}001, \ldots, 5{,}000\}$ (jagged lines), and pdfs used to sample the four components of the parameter vector (smooth lines). The crosses represent the values used in the simulations.

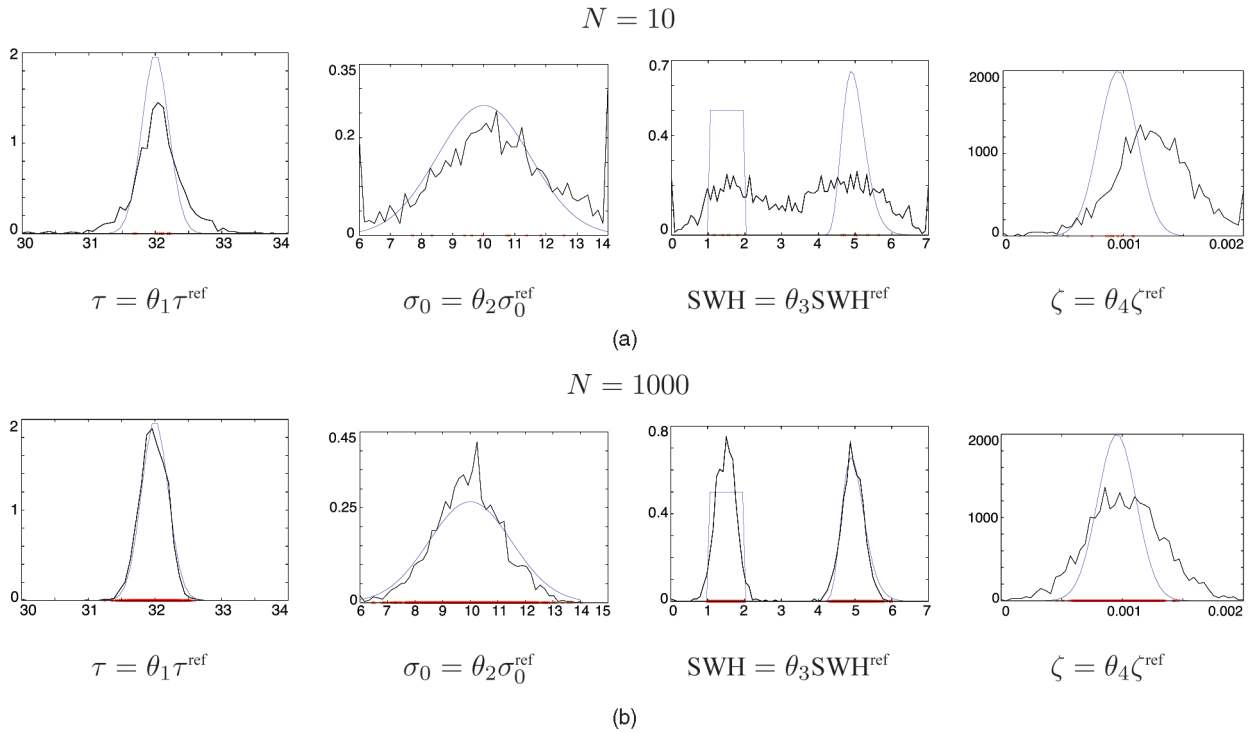

Fig. 6. Histograms of the four components of the parameter vector computed from the samples $\widetilde{\theta}^{(l)}$ for $l \in \{3{,}001, \ldots, 5{,}000\}$ (jagged lines), and pdfs used to sample the four components of the parameter vector (smooth lines). The crosses indicate the values used in the simulations. The number of training data is (a) $N = 10$ and (b) $N = 1{,}000$ (bottom row).

corresponding to the different kinds of the surfaces would considerably simplify this extraction of information. The real data studied in this section are grouped into four classes, referred to as "Ocean," "Ice," "Forest," and "Desert" as described in Table 1. Ten training data from each class are plotted in Fig. 7, where it should be noted that the amplitudes can differ significantly from one class to another, some having a large variability.

In order to assess the learning ability of our classification strategy, $N = 1{,}000$ training data from each of the four classes were selected randomly for training. The MCMC algorithm was run independently on these four data sets, with $\tau^{\text{ref}} = 32$, $\sigma_0^{\text{ref}} = 9$, $\text{SWH}^{\text{ref}} = 2$, and $\zeta^{\text{ref}} = 0.001$. This

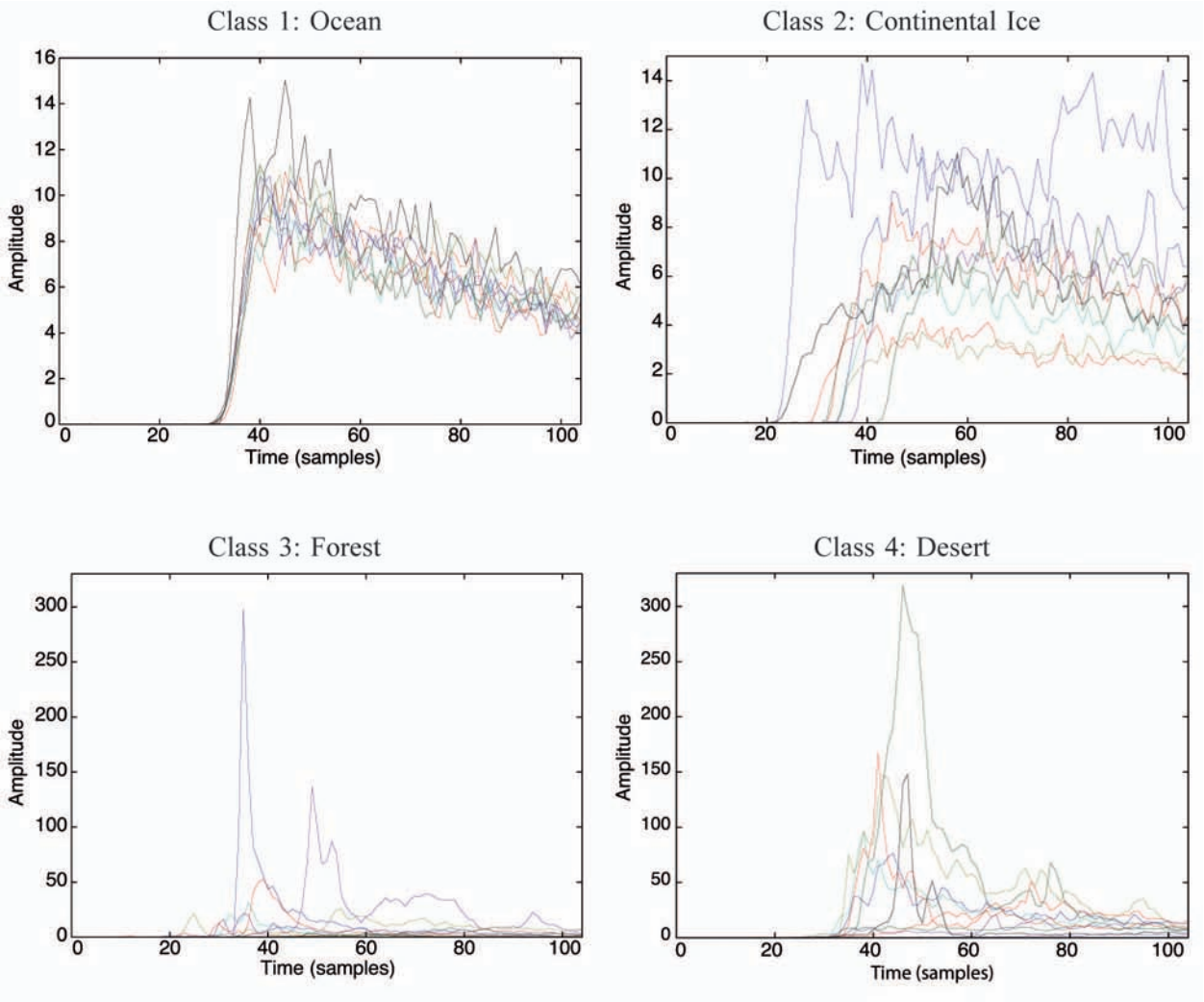

Fig. 7. Ten of the real altimetry time series (i.e., 10 data) used to train the classifier observed over the different surfaces.

procedure was repeated five times by changing the random sample of $N = 1{,}000$ data for each class, leading to the histograms in Fig. 8. As can be seen, the procedure ensures a stable learning of the parameter posterior distribution. Moreover, it can be seen that this distribution is multimodal and/or non-Gaussian in several parameters, which justifies the approach implemented here. Once the model parameters have been estimated conditioned on each class, the estimates can be plugged into the class pdfs following the *plug-in* classification rule [4]. The results obtained with this plug-in rule for the four-class problem defined in Table 1 are depicted in Table 2.

For comparison, the results obtained when using a Gaussian prior for the unknown parameter vector are provided in Table 3. These results have been computed with the same code as in the DPM prior case with the additional constraint $\alpha = 10^{-300}$. The proposed classification rule based on a DPM prior for θ yields better performance than a naive Bayesian rule based on a Gaussian prior for θ, although the latter is already quite sophisticated. The proposed DPM-based classifier was also compared to the one-versus-one and one-versus-all SVM multiclass methods with a polynomial kernel [33, p. 59]. The average of 50 confusion matrices obtained with 1,000 training samples and 1,000 test samples is provided in Tables 4 and 5. The DPM-based classification method shows better performance than the SVM approaches (because it uses the generative Brown model) at the price of a higher computational cost. The advantage of using a generative model allows one to encode the mapping between a reduced set of parameters and the data. An alternative strategy could be to estimate the parameters and thus implement an SVM-based classifier in the space of the parameters. However, this strategy is less robust insofar as it does not capture the uncertainty about the parameters.

TABLE 1
Data Used to Train and Test the Classifier

Class	1: Ocean	2: Ice	3: Forest	4: Desert
Nb. of radar waveforms	9640	10480	8920	11000

6 Conclusions

This paper studied a new supervised classification algorithm based on a Bayesian model whose parameter priors

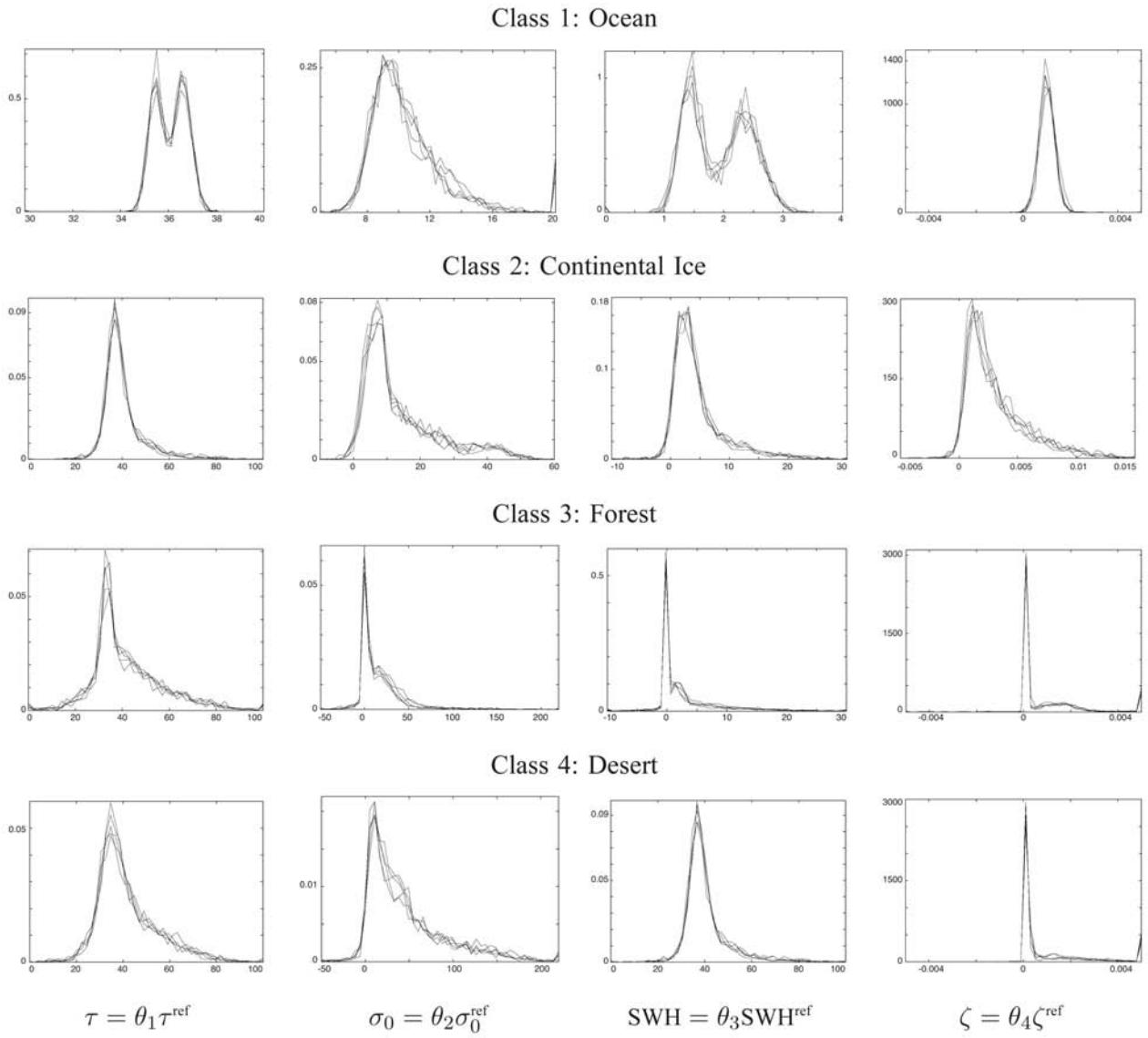

Fig. 8. Histograms of the four components of θ computed from the samples $\widetilde{\theta}^{(l)}$ for $l \in \{3{,}001, \ldots, 5{,}000\}$ for the four classes (marginal distributions). In order to assess the stability of the learning procedure, five different training sets were used, each containing 1,000 training data.

are Dirichlet process mixtures. Note again that Dirichlet process mixtures are interesting since they can be used to describe accurately a large class of probability distributions. An efficient learning algorithm was developed to estimate the class posterior distributions required for Bayesian classification. The resulting classification rule was applied to a practical problem consisting of classifying radar altimeter waveforms backscattered from different surfaces (i.e., oceans, ices, forests, and deserts). The obtained results are promising for this application. A recent study has shown that linear discriminant analysis performed on appropriate features (extracted from the Brown model) can also provide interesting classification results for altimetric signals [24]. A comparison between this classification strategy and the proposed DPM-based classification rule is currently under investigation.

TABLE 2
Confusion Matrix for the Four-Class Altimetry Problem Using a DPM Prior for θ

	Estimated Class: 1	Estimated Class: 2	Estimated Class: 3	Estimated Class: 4
True class: 1	98.71%	1.08%	0.17%	0.04%
Tue class: 2	1.89%	65.79%	18.76%	13.56%
True class: 3	0.44%	12.47%	65.33%	21.76%
True class: 4	0.81%	18.96%	25.05%	55.17%

TABLE 3
Confusion Matrix for the Four-Class Altimetry Problem Using a Gaussian Prior for θ

	Estimated Class: 1	Estimated Class: 2	Estimated Class: 3	Estimated Class: 4
True class: 1	97.08%	1.55%	0.17%	1.20%
True class: 2	8.13%	58.48%	20.03%	13.36%
True class: 3	4.04%	39.69%	35.19%	21.08%
True class: 4	1.93%	29.49%	20.68%	47.89%

TABLE 4
Confusion Matrix for the Four-Class Altimetry Problem Using the One-versus-One SVM Multiclass Method [33, p. 59] (1,000 Training Samples and 1,000 Test Samples)

	Estimated Class: 1	Estimated Class: 2	Estimated Class: 3	Estimated Class: 4
True class: 1	95.91%	2.68%	0.38%	1.03%
True class: 2	34.18%	44.06%	11.12%	10.64%
True class: 3	4.82%	25.79%	44.52%	24.87%
True class: 4	8.23%	27.96%	21.24%	42.57%

TABLE 5
Confusion Matrix for the Four-Class Altimetry Problem Using the One-versus-All SVM Multiclass Method [33, p. 59] (1,000 Training Samples and 1,000 Test Samples)

	Estimated Class: 1	Estimated Class: 2	Estimated Class: 3	Estimated Class: 4
True class: 1	59.76%	31.66%	5.95%	2.63%
True class: 2	18.85%	49.98%	19.52%	11.65%
True class: 3	3.67%	27.74%	47.64%	20.95%
True class: 4	4.34%	26.84%	23.71%	45.11%

APPENDIX A
A GLOSSARY OF NOTATIONS

- J: number of classes
- $C_1, \ldots, C_J$: classes
- $\mathbf{X}_j = \{\mathbf{x}_{1,j}, \ldots, \mathbf{x}_{N_j,j}\}$: training set for class C_j
- $\mathbf{x}_{i,j}$: ith training vector for class C_j
- N_j: number of training vectors for class C_j
- M_j: model for class C_j
- $\theta_{i,j}$: parameter vector associated to the training vector $\mathbf{x}_{i,j}$
- $\boldsymbol{\epsilon}_{i,j}$: noise associated to the training vector $\mathbf{x}_{i,j}$
- $\Theta_j \subset \mathbb{R}^d$: parameter space for class C_j
- d: dimension of the parameter space
- $p_j(\mathbf{x}_{i,j}|\theta_{i,j})$: pdf of $\mathbf{x}_{i,j}$ (belonging to the class C_j) conditioned upon $\theta_{i,j}$
- $p(\theta_j|\mathbf{X}_j)$: parameter posterior distribution for the training set $\mathbf{X}_j$ associated to class C_j
- $p_j(\theta_j|\boldsymbol{\phi}_j)$: prior distribution for θ_j
- $\boldsymbol{\phi}_j$: hyperparameter vector for class C_j
- $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$: generic training set
- $\mathbf{x}$: generic observation vector to be classified
- $p(\mathbf{x}|\mathbf{X}_j)$: predictive density of $\mathbf{x}$ conditioned to the training set $\mathbf{X}_j$
- N: number of training vectors belonging to $\mathbf{X}$
- θ: parameter vector for the generic training set $\mathbf{X}$
- $p(\theta|\mathbf{X})$: parameter posterior distribution for the generic training set $\mathbf{X}$
- $\widetilde{\theta}^{(1)}, \ldots, \widetilde{\theta}^{(L)}$: samples distributed according to $p(\theta|\mathbf{X})$
- $\boldsymbol{\phi}$: hyperparameter vector for the generic training set $\mathbf{X}$
- $p(\boldsymbol{\phi}|\mathbf{X})$: hyperparameter posterior distribution for the generic training set $\mathbf{X}$
- $\widetilde{\boldsymbol{\phi}}^{(1)}, \ldots, \widetilde{\boldsymbol{\phi}}^{(L)}$: samples distributed according to $p(\boldsymbol{\phi}|\mathbf{X})$
- $\mathrm{F} \sim \mathcal{DP}(\alpha, \mathrm{F}_0)$: Dirichlet process with base distribution F_0 and concentration parameter α
- $\omega_1, \omega_2, \ldots$: weights of the Dirichlet process (DP)
- $U_1, U_2, \ldots$: cluster locations of the Dirichlet process
- $\mathcal{B}(\omega; a, b)$: beta distribution with parameters a and b for the random variable ω
- $f(\theta|\boldsymbol{\phi})$: mixture density of a Dirichlet process mixture (DPM)
- $\mathbf{z} = (z_1, \ldots, z_N)$: vector containing the latent variables associated to the vectors $\theta_1, \ldots, \theta_N$ where $\theta_i \sim \mathrm{G}$
- $\mathbf{U} = (U_1, \ldots, U_N)$: cluster locations of $\theta_1, \ldots, \theta_N$ where $\theta_i \sim \mathrm{G}$
- z: latent variable associated to a generic parameter vector θ

- $\mathrm{p}(z|\mathrm{F})$: discrete distribution of the latent variable z associated to the Dirichlet process F
- $\mathrm{G}(\theta)$: DPM probability measure
- $M(t,\theta)$: Model for an altimetric signal with parameter vector θ
- $\mathbf{x} = [\mathbf{x}(1), \ldots, \mathbf{x}(T)]$: time-varying radar waveform
- $\boldsymbol{\epsilon}(t)$: vector containing noise samples for an altimetric signal
- $\mathcal{G}a(\omega; a, b)$: gamma distribution with parameters a and b for the random variable ω
- P_n: known thermal noise level
- ξ_1, ξ_2: known coefficients depending on the kind of altimeter radar
- $\tau, \mathrm{SWH}, \sigma_c, \zeta$: unknown parameters for the altimetric waveform
- $\mathcal{N}(\theta; \mu, \Sigma)$: multivariate normal distribution with mean vector μ and covariance matrix Σ for the random vector θ
- $\mathcal{NIW}(\mu, \Sigma; \mu_0, \kappa_0, \nu_0, \Lambda_0)$: normal-inverse Wishart distribution with parameters $\mu_0, \kappa_0, \nu_0, \Lambda_0$ for the random vector and matrix μ and Σ
- $\mathcal{IW}(\Sigma; \nu_0, \Lambda_0)$: inverse Wishart distribution with parameters ν_0, Λ_0 for the random matrix Σ
- $\mathcal{P}(\nu_0; \lambda_\nu)$: Poisson distribution with parameter λ_ν for the random variable ν_0
- $\mathcal{I}(\mathbf{z})$: set of values taken by the variables contained in $\mathbf{z}$
- $\#\mathcal{I}(\mathbf{z})$: number of elements in $\mathcal{I}(\mathbf{z})$
- μ, Σ: hyperparameters associated to θ for the classification of altimetric signals
- $\mu_0, \kappa_0, \nu_0, \Lambda_0$: base function hyperparameters associated to the hyperparameter prior
- $\mu_N, \kappa_N, \nu_N, \Lambda_N$: parameters of the DPM posterior distribution for the classification of altimetric signals
- $\mathbb{I}_{\{d,d+1,\ldots\}}(.)$: indicator variable for the set (d, ∞)
- $\alpha'_\kappa, \beta'_\kappa, \lambda'_\nu, \nu', \Lambda'^{-1}$: parameters of the hyperparameter posterior distribution for the classification of altimetric signals

APPENDIX B
SAMPLING DPM LATENT VARIABLES AND CLUSTER LOCATIONS

This appendix summarizes three algorithms which can be used to sample DPM latent variables and cluster locations:

- The Algorithm 4 details Step 1.1 of the Gibbs sampler in Algorithm 3 used to sample the DP in the case where the base distribution F_0 is conjugate for the mixture distribution $f(\cdot|\cdot)$, following [34, Algorithm 2].
- Algorithm 5 extends Algorithm 4 to the more general case where F_0 is not conjugate for $f(\cdot|\cdot)$ and is derived from [34, Algorithm 5].
- Algorithm 6 addresses Step 1.2 of Algorithm 3 above in all cases.

Before presenting these algorithms, let us introduce some notations.

Let $\mathcal{I}(\mathbf{z})$ denote the set of values taken by the variables $z_1, \ldots, z_N$. Indeed, for the sake of simplicity, we do not assume that the z_is take all their values in a set $\{1, \ldots, k_{\max}\}$, but instead some values spread among the integers. The reason for this is that the Gibbs samplers may create and suppress locations U_ks, thus incrementing the largest k while forming gaps in-between "alive" indexes k. Thus, $\mathcal{I}(\mathbf{z})$ contains the values of k that are effectively used at a given iteration. We denote $N_{-i,k}(\mathbf{z}) = \sum_{\mathbf{i'}=\mathbf{1},\mathbf{i'}\neq\mathbf{i}}^{\mathbf{N}} \delta_{\mathbf{k},\mathbf{z}_{\mathbf{i'}}}$ the number of variables $z_{i'}$s ($i' \neq i$) which are equal to k.

Algorithm 4. Sampling the latent variables (conjugate case)

- Let $\mathbf{z}' \leftarrow \widetilde{\mathbf{z}}^{(\mathbf{l-1})}$
- For $i = 1, \ldots, N$, do
  - if $z'_i \neq z'_{i'}$ for all $i' \neq i$ (singleton cluster), then remove $\widetilde{U}'^{(l)}_{z'_i}$ from the state.
  - Sample $z'_i \sim \mathrm{p}(z'_i|\mathbf{z}'_{-\mathbf{i}}, \widetilde{\theta}_{\mathbf{i}}^{(\mathbf{l-1})}, \widetilde{\mathbf{U}}^{(\mathbf{l-1})})$ such that

$$\mathrm{p}\big(z'_i = k \big| \mathbf{z}'_{-\mathbf{i}}, \widetilde{\theta}_{\mathbf{i}}^{(\mathbf{l-1})}, \widetilde{\mathbf{U}}^{(\mathbf{l-1})}\big) \propto$$

$$\begin{cases} N_{-i,k} f\big(\widetilde{\theta}_i^{(l-1)} \big| \widetilde{U}_k^{(l-1)}\big), & \text{for } k \in \mathcal{I}(\mathbf{z}'), \\ \alpha \displaystyle\int f\big(\widetilde{\theta}_i^{(l-1)} \big| U\big) \mathrm{F}_0(dU), & \text{for a new } k \notin \mathcal{I}(\mathbf{z}'), \\ 0, & \text{for all other values of } k, \end{cases} \tag{57}$$

  and, if a new cluster is created, sample $\widetilde{U}'^{(l-1)}_{z'_i} \sim p(U|\widetilde{\theta}_i^{(l-1)}) \propto f(\widetilde{\theta}_i^{(l-1)}|U)\mathrm{F}_0(U)$
- Let $\widetilde{z}_i^{(l)} \quad z'_i$

There are many cases where the base distribution is not conjugate for $f(\cdot|\cdot)$ in practical applications. In this case, MH steps are necessary, and they may be implemented as follows (derived from [34, Algorithm 5]):

Algorithm 5. Sampling the latent variables (general case)

- Let $\mathbf{z}' \leftarrow \widetilde{\mathbf{z}}^{(\mathbf{l-1})}$
- For $i = 1, \ldots, N$, do
  - Sample a candidate $z_i^\star$ from the Polya urn probabilities $Q(z_i^\star = k) =$

$$\begin{cases} \frac{N_{-i,k}(\mathbf{z})}{\alpha+N-1}, & \text{for } k \in \mathcal{I}(\mathbf{z}'), \\ \frac{\alpha}{\alpha+N-1}, & \text{for a new } k \notin \mathcal{I}(\mathbf{z}'), \\ 0, & \text{for all other values of } k. \end{cases} \tag{58}$$

  - if $z_i^\star \in \mathcal{I}(\mathbf{z}')$, then compute $r_1 = f(\widetilde{\theta}_i^{(l-1)}|U'_{z_i^\star})/ f(\widetilde{\theta}_i^{(l-1)}|U'_{z'_i})$. With probability $\min(1, r_1)$, set $z'_i \quad z_i^\star$.
  - Otherwise, compute $r_1 = f(\widetilde{\theta}_i^{(l-1)}|U^\star)/f(\widetilde{\theta}_i^{(l-1)}|U'_{z'_i})$, where $U^\star \sim \mathrm{F}_0$. With probability $\min(1, r_1)$, set $z'_i \quad z_i^\star$ and $U'_{z'_i} \leftarrow U^\star$.
  - Let $\widetilde{z}_i^{(n)} \quad z'_i$

Once the latent variables are sampled, the cluster locations can be updated as indicated below.

Algorithm 6. Sampling the cluster locations (all cases)

- Let $\mathbf{U}' \leftarrow \widetilde{\mathbf{U}}^{(\mathbf{l-1})}$
- For $k \in \mathcal{I}(\widetilde{\mathbf{z}}^{(l)})$, sample $\widetilde{U}_k^{(l)}$ from $p(U_k|\widetilde{\theta}_i^{(l-1)}$ with i such that $\widetilde{z}_i^{(l)} = k) \propto \mathrm{F}_0(U_k) \displaystyle\prod_{\substack{i=1,\ldots,N \\ \text{such that } \widetilde{z}_i^{(l)}=k}} f(\widetilde{\theta}_i^{(l-1)}|U_{-k})$, as follows:
  - In the conjugate case, sample directly from $p(U_k|\widetilde{\theta}_i^{(l-1)}$ with i such that $\widetilde{z}_i^{(l)} = k)$
  - In the non-conjugate case, sample $U^\star \sim q(U^\star|U'_k)$, then compute

$$r_2 = \frac{q(U_k'|U^\star)}{q(U^\star|U_k')} \frac{\mathrm{F}_0(U^\star)}{\mathrm{F}_0(U_k')} \prod_{\substack{i=1,\ldots,N \\ \text{such that } \widetilde{z}_i^{(l)}=k}} \frac{f(\widetilde{\theta}_i^{(l-1)}|U^\star)}{f(\widetilde{\theta}_i^{(l-1)}|U_k')}, \qquad (59)$$

and, with probability $\min(1, r_2)$, set $\widetilde{U}_l^{(l)} \quad U^{(\star)}$;
otherwise, set $\widetilde{U}_l^{(l)} \leftarrow U_k'$

The implementation of Algorithm 6 requires sampling from the normal-inverse Wishart distribution $\mathcal{NIW}(\mu, \Sigma; \widetilde{\mu}_N, \widetilde{\kappa}_N, \widetilde{\nu}_N, \widetilde{\Lambda}_N)$, where $\widetilde{\mu}_N$, $\widetilde{\nu}_N$, $\widetilde{\kappa}_N$, and $\widetilde{\Lambda}_N$ are defined as in (34)-(38), with the θ_is being the $\widetilde{\theta}_i^{(l-1)}$s such that $\widetilde{z}_i^{(l)} = k$.

ACKNOWLEDGMENTS

The authors would like to thank L. Amarouche, P. Thibaut, and O.Z. Zanife from *Collecte Localisation Satellites* (CLS), Toulouse, France, for providing the real data sets and for fruitful discussions regarding altimetry.

REFERENCES

[1] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.
[2] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. Wiley, 2001.
[3] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.
[4] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, Jan. 2000.
[5] S. Theodoridis and K. Koutroubas, *Pattern Recognition.* Academic Press, 1999.
[6] R. Kass and A. Raftery, "Bayes Factors," *J. Am. Statistical Assoc.,* vol. 90, no. 430, pp. 773-795, 1995.
[7] R. Raina, Y. Shen, A. Ng, and A. McCallum, "Classification with Hybrid Generative/Discriminative Models," *Advances in Neural Information Processing Systems,* vol. 16, MIT Press, 2004.
[8] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *Proc. Nat'l Conf. Artificial Intelligence Workshop Learning for Text Categorization,* vol. 752, 1998.
[9] H. Schneiderman and T. Kanade, "A Statistical Approach to 3D Object Detection Applied to Faces and Cars," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 746-751, June 2000.
[10] I. Kononenko, "A Semi-Naive Bayesian Classifier," *Proc. Sixth European Working Session on Learning,* pp. 206-219, Mar. 1991.
[11] Y. Teh, M. Jordan, M. Beal, and D. Blei, "Hierarchical Dirichlet Processes," *J. Am. Statistical Assoc.,* vol. 101, no. 476, pp. 1566-1581, 2006.
[12] D. Blei and M. Jordan, "Variational Inference for Dirichlet Process Mixtures," *J. Bayesian Analysis,* vol. 1, no. 1, pp. 121-144, 2005.
[13] P. Muller and F. Quintana, "Nonparametric Bayesian Data Analysis," *Statistical Science,* vol. 19, no. 1, pp. 95-110, 2004.
[14] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian Inference for Linear Dynamic Models with Dirichlet Process Mixtures," *IEEE Trans. Signal Processing,* vol. 56, no. 1, pp. 71-84, Jan. 2008.
[15] E. Jackson, M. Davy, A. Doucet, and W. Fitzgerald, "Bayesian Unsupervised Signal Classification by Dirichlet Process Mixtures of Gaussian Processes," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* Apr. 2007.
[16] M. Davy, C. Doncarli, and J.Y. Tourneret, "Classification of Chirp Signals Using Hierarchical Bayesian Learning and MCMC Methods," *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 377-388, Feb. 2002.
[17] C. Antoniak, "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems," *The Annals of Statistics,* vol. 2, pp. 1152-1174, 1974.
[18] B. Shahbaba and R.M. Neal, "Nonlinear Models Using Dirichlet Process Mixtures," Technical Report 0707, Dept. of Statistics, Univ. of Toronto, 2007.
[19] T. Ferguson, "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics,* vol. 1, pp. 209-230, 1973.
[20] J. Sethuraman, "A Constructive Definition of Dirichlet Priors," *Statistica Sinica,* vol. 4, pp. 639-650, 1994.
[21] M. Escobar and M. West, "Bayesian Density Estimation and Inference Using Mixtures," *J. Am. Statistical Assoc.,* vol. 90, pp. 577-588, 1995.
[22] D. Blackwell and J. MacQueen, "Ferguson Distributions via Polya Urn Schemes," *The Annals of Statistics,* vol. 1, no. 2, pp. 353-355, 1973.
[23] S. Richardson and P.J. Green, "On Bayesian Analysis of Mixtures with an Unknown Number of Components," *J. Royal Statistical Soc. Series B,* vol. 59, no. 4, pp. 731-792, 1997.
[24] J.-Y. Tourneret, C. Mailhes, L. Amarouche, and N. Steunou, "Classification of Altimetric Signals Using Linear Discriminant Analysis," *Proc. IEEE Int'l Geoscience and Remote Sensing Symp.,* July 2008.
[25] C.J. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images.* Artech House, 1998.
[26] J. Severini, C. Mailhes, P. Thibault, and J.-Y. Tourneret, "Bayesian Estimation of Altimeter Echo Parameters," *Proc. IEEE Int'l Geoscience and Remote Sensing Symp.,* July 2008.
[27] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin, *Bayesian Data Analysis,* second ed., Chapman & Hall/CRC, 2004.
[28] G. Brown, "The Average Impulse Response of a Rough Surface and Its Applications," *IEEE Trans. Antennas and Propagation,* vol. 25, no. 1, pp. 67-74, Jan. 1977.
[29] S. Baker, O. Bombaci, C. Zeli, P. Venditti, O.-Z. Zanife, B. Soussi, J.-P. Dumont, J. Stum, M.P. Milagro-Perez, and J. Benveniste, "ENVISAT RA-2/MWR Product Handbook," http://wwwcpg.mssl.ucl.ac.uk/RA2_Handbook/concepts/ra2/ra2-mwr-PH-2.7.1.html#pgfId-868561, Mar. 2002.
[30] J.P. Dumont, "ALT-RET-OCE-02—to Perform the Ocean-2 Retracking—Definition, Accuracy and Specification," technical report, Collecte Localisation Satellite (CLS), Toulouse, France, Oct. 2001.
[31] C. Fraley and A.E. Raftery, "Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering," Technical Report 05/486, Dept. of Statistics, Univ. of Washington, Seattle, Aug. 2005.
[32] T.W. Anderson, *An Introduction to Multivariate Statistical Analysis,* third ed. Wiley, 2003.
[33] R. Herbrich, *Learning Kernel Classifiers.* MIT Press, 2002.
[34] R. Neal, "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," Technical Report 9815, Dept. of Statistics and Dept. of Computer Science, Univ. of Toronto, Ontario, Canada, 1998.

Manuel Davy received the engineer degree in electrical engineering from the Ecole Centrale de Nantes, France, in 1996, and the PhD degree from the University of Nantes in 2000. From 2000 to 2002, he was a research associate in the Signal Processing Group, University of Cambridge. In 2007, he created VEKIA in Lille, France, a company dedicated to software edition for forecasting solutions in retail. He is a member of the IEEE.

Jean-Yves Tourneret received the ingénieur degree in electrical engineering from the Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique et d'Hydraulique in Toulouse (ENSEEIHT) in 1989 and the PhD degree from the National Polytechnic Institute, Toulouse in 1992. He is currently a professor at the University of Toulouse, France (ENSEEIHT) and a member of the IRIT Laboratory (UMR 5505 of the CNRS). His research activities are centered around statistical signal processing, with a particular interest in classification and Markov Chain Monte Carlo methods. He was the program chair of the European Conference on Signal Processing (EUSIPCO), which was held in Toulouse in 2002. He was also member of the organizing committee for the international conference ICASSP '06 which was held in Toulouse in 2006. He has been a member of different technical committees. He is currently serving as an associate editor for the *IEEE Transactions on Signal Processing*. He is a senior member of the IEEE.