NONPARAMETRIC BAYESIAN SUPERVISED CLASSIFICATION OF FUNCTIONAL DATA

Asma Rabaoui[1]*, Hachem Kadri*[2]*, Manuel Davy*[3]

[1]LAPS-IMS/CNRS, Université de Bordeaux, Talence, France
[2]SequeL Project, INRIA Lille - Nord Europe, Villeneuve d'Ascq, France
[3]CNRS/LAGIS/Vekia SAS, Ecole Centrale de Lille, Villeneuve d'Ascq, France

ABSTRACT

A nonparametric approach combining generative models and functional data analysis is presented in this paper for classifying functional data which arise naturally in a wide variety of signal processing applications, such as brain computer interfacing, speech recognition, or image classification. Based on a new and improved family of Bayesian classifiers, we extend hierarchical Bayesian classification methodology from vector to functional settings. We provide theoretical and practical motivations to our approach which relies on Dirichlet process mixtures and Gaussian processes. The performance is evaluated on phoneme recognition task, and compared to that of Functional Support Vector Machines (FSVMs).

Index Terms— Functional data analysis, supervised classification, Dirichlet process mixtures, Gaussian processes, MCMC.

1. INTRODUCTION

In many Signal Processing applications, the data are collected by sampling a random process realization that can be seen as a continuous function. One may think about such functions as defined over time or/and space. Examples are electroencephalogram (EEG) signals in Brain Computer Interfacing (BCI) [1], pitch contours in speech synthesis [2], or functional magnetic resonance images (fMRI) in neuroimaging [3]. Functional data analysis (FDA) is a very attractive field of research that provides the possibility to fully exploit specific properties of the functions that describe these continuous data. For an introduction to relevant concepts and potential applications of functional data analysis, we refer the reader to the the book by Ramsay and Silverman [4].

There is clearly a number of compelling reasons for developing a classification methodology in the functional data framework instead of applying multivariate analysis; most importantly: (1) taking into account relationships between samples and the functional nature of data (e.g. smoothness of the curves underlying the discrete observations) ; (2) dealing with the case where the observation locations (or data acquisition intervals) are different from one curve to another ; (3) controlling and reducing the errors in the measurements (noisy observations).

The idea developed in this paper is to extend existing functional analysis methods to address the supervised classification problem, when a set of training data is available. This concerns a number of applications such as listed above, and is therefore a key task in many Signal Processing problems. In a recent work [5], the authors develop a generative classifier for vectorial data based on hierarchical Bayesian models and Markov chain Monte Carlo (MCMC). In this work, the model parameter prior distribution is assumed to be a mixture of Dirichlet processes (DPM) which have the advantage of describing accurately a large class of probability distributions. In [5], it is shown that the learning algorithm performance is high, and outperforms SVMs, in classifying altimetric satellite measurement signals.

A primary motivation for our paper is to build on this previous study and extend the DPM-Bayesian supervised classification method to the case where input data (signals) are functions. Contrary to [5] in which data are considered to be vectors, this paper assumes them to be realizations of Gaussian Processes (GP) so as to take into account the inherent functional nature of the data. It is interesting to mention here that though some concepts on the use of Dirichlet process mixtures with Gaussian processes are proposed in [6], the present work addresses supervised classification rather than clustering, and use a DPM as a prior over the GP parameters.

A key feature of the proposed framework is that when no precise generative data model can be defined, the GP model can be used as a powerful modeling alternative, as the smoothness of the data can be controlled by selecting a given covariance function to define the GP. In other words, the proposed framework assumes that each data sequence in the training set is sampled from a GP realization, each class being characterized by the statistics of the GP underlying each sequence of data. Its interest is that is quite flexible, and can be applied in a number of applications.

The remainder of this paper is organized as follows. In Section 2, we present the proposed functional classifier based on Gaussian processes and Dirichlet process mixtures. We illustrate the performance of our approach through phoneme recognition experiments in Section 3. Section 4 provides some conclusions and future work directions.

2. SUPERVISED FUNCTIONAL CLASSIFIER

Gaussian processes have received substantial attention in machine learning and statistics communities in recent years [7, 8]. They provide flexible tools for various problems, such as Bayesian regression. More recently, a Bayesian Dirichlet process mixtures of Gaussian processes was proposed in [6] to deal with the clustering problem in unsupervised settings. In this section we show how to build on it a new supervised functional classifier, by merging with the approach in [5]. To represent observed data as functions, we consider each input signal, denoted $\mathbf{x}(\cdot)$, to be a realization of a Gaussian process.

2.1. Methodology

Consider a supervised classification problem of functional data with K classes denoted as $C_1, ..., C_K$, each containing a set of training data (functions) $\mathbf{X}_k = \{\mathbf{x}_{1,k}(\cdot), \ldots, \mathbf{x}_{N_k,k}(\cdot)\}$, $k = 1, \ldots, K$. These functions are assumed to relied on a covariate $t \in \mathbb{R}^d$, while the function itself takes its values in $\mathbb{R}$. For time series, $d = 1$ and t is times while for images, $d = 2$ and t are the coordinates in the image plane.

A practically important assumption is that a continuous signal $\mathbf{x}_{i,k}(\cdot)$ is known through a set of observed points $\{\mathbf{x}_{i,k}(t_p^{i,k}), p = 1, \cdots, T^{i,k}\} \in \mathbb{R}$ where $t_p^{i,k} \in \mathbb{R}^d$. Note that neither the sampling coordinates $t^{i,k}$'s nor the number of samples $T^{i,k}$'s are assumed to be the same for all the signals.

Modeling functional data with Gaussian processes. We assume each $\mathbf{x}_{i,k}(\cdot)$, $i = 1, \ldots, N_K$ is a realization of a Gaussian process (GP) and has the following generative model

$$\mathbf{x}_{i,k}(\cdot) \sim \mathcal{GP}(\mathbf{x}_{i,k}(\cdot); \mathbf{m}_{i,k}(\cdot), \mathbf{K}_{i,k}(\cdot,\cdot)) \tag{1}$$

where $\sim$ means "distributed as". A Gaussian process is characterized by its mean function $\mathbf{m}_{i,k}(\cdot) : \mathbb{R}^d \to \mathbb{R}$ and its covariance function $\mathbf{K}_{i,k}(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and its outcomes are functions. Following the work of Shi [8], we will assume that the functions $\mathbf{x}_{i,k}(\cdot)$ are generated from zero-mean GPs, that is, $\mathbf{m}_{i,k}(\cdot) \equiv 0$, and we consider covariance functions having the form

$$\begin{aligned}\mathbf{K}_{i,k}(s,t;\theta_{i,k}) = a_0 + a_1 \sum_{q=1}^{d} s_q t_q \\ + a_2 \exp\left(-\frac{1}{2}\sum_{q=1}^{d} b_q |s_q - t_q|^2\right)\end{aligned} \tag{2}$$

with $(d+3)$-dimensional parameter $\theta_{i,k} = \{a_0, a_1, a_2, b_1, \cdots, b_d\}$, for $(s,t) \in \mathbb{R}^d \times \mathbb{R}^d$. The parameters $\theta_{i,k}$ belong to a space Θ_k, $i = 1, \ldots, N_k$. Note that we will not focus in this paper on this specific choice of the covariance function, the methodology presented here remains valid for other forms than (2). In the following, to keep the notations short, we use θ_k to denote the set of parameters $\{\theta_{1,k}, \cdots, \theta_{i,k}, \cdots, \theta_{N_k,k}\}$ for class $\#k$.

Bayesian supervised classification framework. Let $\mathsf{L}_k(\mathbf{x}_k|\theta_k)$ denote the likelihood of observing $\mathbf{x}_k$, assumed to belong to class C_k. If we assume that $\mathbf{x}_k$ is observed through a set of T samples (therefore, a T-dimensional vector), then its likelihood is a T-dimensional multivariate normal distribution $\mathcal{N}_T$ which satisfies the following relation

$$\begin{aligned}\mathsf{L}(\mathbf{x}_k|\theta_k) &= \mathcal{N}_T(\mathbf{x}_k;\theta_k) \\ &\propto (2\pi)^{-\frac{T}{2}} |\mathbf{K}(\cdot,\cdot;\theta_k)|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k^T \mathbf{K}(\cdot,\cdot;\theta_k)^{-1}\mathbf{x}_k)\right]\end{aligned} \tag{3}$$

where $|\mathbf{K}(\cdot,\cdot;\theta_k)|$ denotes the determinant of $\mathbf{K}(\cdot,\cdot;\theta_k)$ and $\propto$ stands for "is proportional to". Now, following the bayesian methodology, let $p(\theta_k|\mathbf{X}_k)$ be the posterior probability density function (pdf) for θ_k given the training data set $\mathbf{X}_k$ of class C_k. A new observation function $\mathbf{x}$ can be classified thanks to the predictive pdf $p(\mathbf{x}|\mathbf{X}_k)$, $k = 1, \ldots, K$ as follows

$$p(\mathbf{x}|\mathbf{X}_k) = \int \mathsf{L}_k(\mathbf{x}|\theta_k) p(\theta_k|\mathbf{X}_k) d\theta_k \tag{4}$$

By assuming the classes have the same prior probabilites of occurence, the Bayesian maximum *a posteriori* (MAP) classifier assigns $\mathbf{x}$ to the class $\widehat{k} = \arg\max_k p(\mathbf{x}|\mathbf{X}_k)$.

To compute $p(\mathbf{x}|\mathbf{X}_k)$ for a given class C_k, the distribution $p(\theta_k|\mathbf{X}_k)$ has to be estimated: this is the aim of the training phase in Bayesian supervised classification. In this work, a nonparametric hierarchical model is used to the model each signal parameters $\theta_{i,k}$ pdf, yielding $p(\theta_{i,k}|\boldsymbol{\phi}_k)$ where $\boldsymbol{\phi}_k$ is a set of hyperparameters with prior pdf $p(\boldsymbol{\phi}_k)$ (see, e.g., [5] and references therein for details). The posterior pdf of the class parameters θ_k is given by

$$\begin{aligned}p(\theta_k|\mathbf{X}_k) &= \int p(\theta_k, \boldsymbol{\phi}_k|\mathbf{X}_k) d\boldsymbol{\phi}_k \\ &= \int p(\theta_k|\boldsymbol{\phi}_k) p(\boldsymbol{\phi}_k|\mathbf{X}_k) d\boldsymbol{\phi}_k\end{aligned} \tag{5}$$

where

$$\begin{aligned}p(\boldsymbol{\phi}_k|\mathbf{X}_k) &= \int \ldots \int p(\theta_{1,k}, \ldots, \theta_{N_k,k}, \boldsymbol{\phi}_j|\mathbf{X}_k) \\ &\qquad d\theta_{1,k} \ldots d\theta_{N_k,k} \\ &\propto \int \ldots \int p(\boldsymbol{\phi}_k) \prod_{i=1}^{N_k} [\mathsf{L}_k(\mathbf{x}_{i,k}|\theta_{i,k}) p(\theta_{i,k}|\boldsymbol{\phi}_k)] \\ &\qquad d\theta_{1,k} \ldots d\theta_{N_k,k}\end{aligned}$$

To sum up, in this work we use a parametric model the GP covariance $\mathbf{K}_{i,k}(\cdot,\cdot)$, then we adopt the bayesian framework and place a prior directly on the parameters of $\mathbf{K}_{i,k}(\cdot,\cdot)$. As demonstrated in [5], Dirichlet Process Mixture (DPM) are suitable tools to model prior knowledge over parameters. This allows for a flexible nonparametric modeling framework. Therefore, in the following, $p(\theta_k|\boldsymbol{\phi}_k)$ is assumed to be a DPM.

Hierarchical DPM prior for the model parameters. A DPM model can basically be thought of as a simple mixture model given by the mixed pdf $\theta_{i,k} \sim p(\theta_{i,k}|\boldsymbol{\phi}_{i,k})$ and prior $\boldsymbol{\phi}_{i,k} \sim \mathbb{G}(\boldsymbol{\phi}_{i,k})$ where $\mathbb{G}$ itself is the random outcome of a Dirichlet Process $\mathcal{DP}(\mathbb{G}_0(\varphi_k), \alpha)$ (that is, a probabilistic distribution over probabilistif distributions). In summary, we have :

$$\begin{aligned}\theta_{i,k}|\boldsymbol{\phi}_{i,k} &\sim p(\theta_{i,k}|\boldsymbol{\phi}_{i,k}) \\ \boldsymbol{\phi}_{i,k}|\mathbb{G} &\sim \mathbb{G}(\cdot) \\ \mathbb{G}|\psi_k &\sim \mathcal{DP}(\mathbb{G}_0(\varphi_k), \alpha)\end{aligned}$$

where $\psi_k = \{\alpha, \varphi_k\}$ is the hyperparameter vector, with φ_k a given parameter vector for $\mathbb{G}_0$. A full definition of this model can be found in [6]. This model can be re-written by introducing a probability density function (pdf) F according to which each $\theta_{i,k}$ is distributed from, $\theta_{i,k} \sim F(.)$. The pdf F is defined by marginilising out $\mathbb{G}$ as follows:

$$F(\theta_{i,k}) = \int_{\boldsymbol{\phi}_{i,k}} p(\theta_{i,k}|\boldsymbol{\phi}_{i,k}) d\mathbb{G}(\boldsymbol{\phi}_{i,k}) \tag{6}$$

In our case, for the sake of generality, the mixed distribution is assumed to be a $(d+3)$-dimensional multivariate Gaussian distribution

$$p(\theta_{i,k}|\boldsymbol{\phi}_{i,k}) = \mathcal{N}_{(d+3)}(\theta_{i,k}; \mu, \Sigma) \tag{7}$$

where $\boldsymbol{\phi}_{i,k} = \{\mu, \Sigma\}$ contains the hyperparameters (μ is the mean vector and Σ is the diagonal covariance matrix). This DPM prior model yields a very large family of pdfs over parameters $\theta_{i,k}$

Hyperparameter prior using DP. We use the DP prior to ensure reliable statistical inference for the hyperparameters of interest, that is $\boldsymbol{\phi}_k$. The advantage of applying the DP prior to hierarchical models has been addressed extensively in the statistics literature, mostly in recent years, see for example [9]. Note that mixture models using a DP as a prior have become increasingly popular for modeling when conventional parametric models would impose unreasonably stiff constraints on the distributional assumptions.

By integrating over $\mathbb{G}$ through the so called polya urn representation, we see that the joint distribution of $\boldsymbol{\phi}_k = \{\boldsymbol{\phi}_{1,k}, \cdots, \boldsymbol{\phi}_{N_k,k}\}$

may be factored into a product of successive conditional distributions of the following form:

$$\boldsymbol{\phi}_{i,k}|\boldsymbol{\phi}_{-i,k},\psi_k \sim \frac{1}{\alpha+N_k-1}\sum_{j=1,j\neq i}^{N_k}\delta_{\boldsymbol{\phi}_{j,k}}+\frac{\alpha}{\alpha+N_k-1}\mathbb{G}_0 \quad (8)$$

where $\boldsymbol{\phi}_{-i,k}$ denotes $\boldsymbol{\phi}_k\backslash\boldsymbol{\phi}_{i,k}$. This factorization implies that $\boldsymbol{\phi}_{i,k}$ has discrete, though infinite, support. This implies that a random draw from $\mathbb{G}$ either equals one of the previous draws or is drawn independently from the base probability measure $\mathbb{G}_0$.

Note that $\mathcal{DP}(\mathbb{G}_0(\varphi_k),\alpha)$ may be represented in an equivalent way by its latent variables and cluster locations ($\mathbf{U}$, $\mathbf{z}$). Therefore, in the following we introduce the notion of a class label z_k assigned to each hyperparameter $\boldsymbol{\phi}_k$. We set $\mathbf{z}=\{z_k,k=1,\cdots,N\}$ and we denote by $\mathcal{I}(\mathbf{z})$ the set of values taken by the labels. The location variables are denoted $\mathbf{U}=\{\mathbf{U}_l,l=1,\cdots,L\}$ such that $\boldsymbol{\phi}_k=\mathbf{U}_{z_k}$. Here, the $\mathbf{U}_l$'s give the model hyperparameters, while z_k's indicates their corresponding labels, such that $z_k=l$.

From the polya urn probabilities in (8), we can claim that the draws for the indicator variables are obtained according to the following sampling scheme:

$$z_k \sim \Pr(z_k|\mathbb{G}) = \begin{cases} \frac{N_{-k,l}(\mathbf{z})}{\alpha+N-1} \text{ for } l\in\mathcal{I}(\mathbf{z}) \\ \frac{\alpha}{\alpha+N-1} \text{ for a new } l\in\mathcal{I}(\mathbf{z}) \end{cases} \quad (9)$$

where $N_{-k,l}(\mathbf{z})=\sum_{k'=1,k'\neq k}^{N}\delta_{l,z'_k}$, the number of z'_k's ($k'\neq k$) which equal l.

For computational reasons, we assume for $\mathbb{G}$ a conjugate normal inverse Wishart base distribution as in [10],

$$\begin{aligned}\mathbb{G}_0(\varphi_k) &= \mathcal{NIW}(\mu,\Sigma;\mu_0,\kappa_0,\nu_0,\Lambda_0),\\ &= \mathcal{N}(\mu;\mu_0,\Sigma/\kappa_0)\mathcal{IW}(\Sigma;\nu_0,\Lambda_0),\end{aligned}$$

where $\mu_0,\kappa_0,\nu_0,\Lambda_0$ are the base function hyperparameters (to which might be added the DPM hyperparameter α) and $\mathcal{IW}(\Sigma;\nu_0,\Lambda_0)$ is the inverse Wishart distribution.

2.2. Efficient sampling

Since it is generally not easy to derive an exact analytical expression for computing the posterior distribution $p(\theta_k|\mathbf{X}_k)$ in (5), the use of MCMC methods permits to generate a Markov chain whose limiting distribution is equal to the desired target distribution. Typically the full conditionals can be sampled from using Gibbs or possibly Metropolis-Hastings updates [11].

Assume a set of samples $\widetilde{\theta}_k^{(n)}$, $n=1,\ldots,N_{\text{iter}}$, distributed according to $p(\theta_k|\mathbf{X}_k)$ (defined in (5)) is available. Then, the integral in Eq.(4) can be estimated as follows

$$p(\mathbf{x}|\mathbf{X}_k) \;=\; \int \mathsf{L}_k(\mathbf{x}|\theta_k)p(\theta_k|\mathbf{X}_k)d\theta_k \;\approx\; \frac{1}{N_{\text{iter}}}\sum_{n=1}^{N_{\text{iter}}}\mathsf{L}_k(\mathbf{x}|\widetilde{\theta}_k^{(n)}) \quad (10)$$

A standard way of generating samples $\widetilde{\theta}_k^{(n)}$ distributed according to $p(\theta_k|\mathbf{X}_k)$ is provided in Algorithm 1.

In the following, note that the class subscript k will be omitted for notational clarity. Given N signals $\mathbf{x}_1,\ldots,\mathbf{x}_N$ and the parametric model in eq.(1)-(2), we aim at estimating the class labels $\mathbf{z}=\{z_k\}_{k=1,\ldots,N}$ as well as the cluster locations $\mathbf{U}$. In the following, we derive the posterior parameter distribution formulated above. Building on [6, 5] yields the following sampling strategy referred to as Algorithm 1.

Algorithm 1: Sampling from the class posterior distribution

% Step 0: Initialization

- Set α to a high fixed value α_0
- 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 $n=1,\ldots,N_{\text{iter}}$, do

1.0- Update the precision parameter as : $\alpha_n=\alpha_{n-1}+\eta(\tau-\alpha_{n-1})$

1.1- Sample $\widetilde{\mathbf{z}}^{(n)} \sim \mathsf{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(n-1)},\widetilde{\boldsymbol{\theta}}^{(n-1)})$ (more details are in [5, pp.1793])

1.2- Sample $\widetilde{\mathbf{U}}^{(n)} \sim p(\mathbf{U}|\widetilde{\mathbf{z}}^{(n)},\widetilde{\boldsymbol{\theta}}^{(n-1)})$ (more details are in [5, pp.1793])

1.3- Sample $\widetilde{\boldsymbol{\theta}}^{(n)} \sim p(\boldsymbol{\theta}|\widetilde{\mathbf{z}}^{(n)},\widetilde{\mathbf{U}}^{(n)})$: for $i=1,\ldots,N$, sample $\widetilde{\theta}_i^{(n)} \sim p(\theta_i|\mathbf{x}_i,\widetilde{U}^{(n)}_{\widetilde{z}_i^{(n)}}) \propto \mathsf{L}(\mathbf{x}_i|\theta_i)f(\theta_i|\widetilde{U}^{(n)}_{\widetilde{z}_i^{(n)}})$ using, e.g, a MH step

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

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

2.3. Discussion

Whilst MCMC provides a convenient way to draw inference from complicated statistical models, there are many, perhaps under appreciated, problems associated with the MCMC analysis of mixtures. the problems are mainly caused by the non-identifiability of the components under symmetric priors, which leads to so called *label switching* in the MCMC output. This will means that ergodic averages of components specific quantities will be identical and thus useless for inference. For a review of some solutions to the label switching problem, such as artificial identifiability constraints and relabeling algorithms, see [12] and references wherein.

In this paper, we define a prior over the hyperparameter α. The convergence of the above algorithm is made easier by initializing α at a large value and letting it decrease through an annealing (tempering) scheme. The initial annealing parameter will be set to an initial fixed value α_0 and then it will be geometrically updated according to $\alpha_l=\alpha_{l-1}+\eta(\tau-\alpha_{l-1})$, at each time step (the parameter, η, can be set to $1/100$ for example).

Note that α is the prior parameter on the number of components in the mixture. By decreasing α we construct a sequence of annealed target posterior distributions. This process leads to a sequence of intermediate distributions that help in exploring the high probability regions of the target posterior. Note that as $l\to\infty$, $\alpha_l\to\tau$ that ensures convergence to the true target posterior. More precisely, α plays the role of a tempering parameter. As this tempering parameter is reduced, the distributions become sharply peaked at the global maxima of the target distribution. However, it is important to mention that there is no guarantee that the samples are distributed according to the target distribution at iteration t all the more since α may not be decreased slowly enough. To make sure the MCMC

algorithm converges to the desired target distribution as the number of simulated annealing runs grows, a Sequential Monte Carlo Sampler can be applied in the spirit of [13]. In this procedure, several Markov chains are simulated in parallel and are assigned weights to correct for the discrepancy between the sampling distribution and the desired posterior distribution.

3. EXPERIMENTS

In this section, we first assess the performance of our supervised functional classification algorithm on synthetic data. Then, we provide some results obtained with phoneme recognition data, where functional data (and more specifically curves data) occur naturally. The aim of the experimental evaluation is twofold. First to illustrate the potential of adopting a nonparametric Bayesian approach for functional data, and second to inspect how effective functional classification under our framework is, compared to functional SVMs [14].

3.1. Synthetic data

The data tested here are generated using GPs with known parameters. The assumptions concerning the model hyperparameters are the same as in [5, pp.1787]. We set $d = 1$ and $\alpha_0 = 20$ and in order to check that the algorithm accurately learns the values of the different model parameters for each training data set, we used Monte Carlo runs that consist of 1000 iterations (including 400 burn-in iterations). Then, in order to assess the learning ability of our classifier, we used data grouped into 2 classes, referred to as C_1 and C_2. We set $a_0 = a_1 = 4$ for both classes and we set $a_2 = 20, b_1 = 15$ for C_1 and $a_2 = 5, b_1 = 9$ for C_2. It should be noted that the amplitudes of the generated signals show some similarities from one class to another. Signals were selected randomly for training (30 signals for each class) and testing (20 signals for each class) steps, and the MCMC algorithm was run independently on these two data sets. 1-dimensional signals are considered with different sampling coordinates. To make decision about the class of each testing signal, the MAP classifier assigns each signal to the class maximizing its predictive pdf as in (4). The classification results are reported in Table 1.

3.2. Phoneme recognition data[1]

The data were extracted from the TIMIT database which is a widely used resource for research in speech recognition, and consist of 4509 log-periodograms corresponding to recording phonemes of 32 ms duration. Following [14], we consider only the most difficult subproblem in the database which consists in classifying the phoneme "aa" in "dark" against "ao" in "water". The database is a multi-speaker database. There are 325 speakers in the training set and 112 in the test set. We have 519 examples for "aa in the training set (759 for "ao) and 176 in the test set (263 for "ao). Table 2 reports the classification error rate obtained using Functional SVMs and the proposed DPM-based functional approach. Functional SVMs results are reproduced from [14]. In the phoneme recognition data, we obtain satisfactory results; the performance of our functional generative classification method is equivalent, or slightly better than, functional SVMs performance.

[1] http://stat.stanford.edu/ tibs/ElemStatLearn/datasets/phoneme.data

4. CONCLUSION

We have proposed a supervised, nonparametric Bayesian approach for classifying functional data, where observed continuous signals are expressed by Gaussian processes and the parameter priors by Dirichlet process mixtures. Experiments have shown that the proposed algorithm achieves good classification results and performs slightly better than Functional SVMs in the phoneme recognition data. However, a more extensive benchmark study remains to be pursued. In future we will explore more experiments, not only on functional datasets but also on time-series and longitudinal datasets, and compare our DPM-based functional approach with other previous discriminative and generative functional methods, such as those in [15, 16].

Table 1. Confusion Matrix for the two class classification problem using the proposed algorithm.

	Estimated Class: 1	Estimated Class: 2
True class: 1	85%	15%
Tue class: 2	10%	90%

Table 2. Classification error rate for Functional SVMs and our generative functional method on the phoneme recognition data.

Functional SVMs	DPM-based functional classifier
19.4%	17.8%

5. REFERENCES

[1] J.P. Thivierge, "Functional data analysis of cognitive events in EEG," in *Proceedings of the IEEE International SMC'07*, October 7-10, 2007, pp. 2473–2478.

[2] M. Gubian, L. Boves, and F. Cangemi, "Joint analysis of f0 and speech rate with functional data analysis," in *ICASSP'11*, May 22-27, 2011, pp. 4972–4975.

[3] V. Solo, P. Purdon, and E. Brown, "Spatio-temporal signal processing for multi-subject functional MRI studies," in *ICASSP'01*, May 7-11, 2001, pp. 3441–3444.

[4] J.O. Ramsay and B.W. Silverman, *Functional Data Analysis, 2nd ed*, Springer Verlag, New York, 2005.

[5] M. Davy and J.Y. Tourneret, "Generative supervised classification using dirichlet process priors," *IEEE Trans. PAMI*, vol. 32, no. 10, pp. 1781–1794, 2010.

[6] E. Jackson, M. Davy, A. Doucet, and W. Fitzgerald, "Bayesian unsupervised signal classification by dirichlet process mixtures of gaussian processes," in *ICASSP'07*, April 15-20, 2007, pp. 1077–1080.

[7] C.E. Rasmussen, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71, 2006.

[8] J.Q. Shi, R. Murray-Smith, and M. Titterington, "Hierarchical Gaussian process mixtures for regression," *Statistics and Computing*, vol. 15, pp. 31–41, 2005.

[9] R.M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 249–265, 2000.

[10] C. Fraley, "Bayesian regularization for normal mixture estimation and model-based clustering," *Journal of classification*, vol. 220, no. 24, pp. 155–181, 2007.

[11] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, New York: Springer-Verlag : Second edition, 2004.

[12] A. Jasra, C. Holmes, and D. Stephens, "Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling," *Statistical Science*, vol. 20, no. 1, pp. 50–67, 2001.

[13] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *Journal of the Royal Statistical Society, Series B*, vol. 68, no. 3, pp. 411–436, 2006.

[14] Fabrice Rossi and Nathalie Villa, "Support vector machine for functional data classification," *Neurocomputing*, vol. 69, no. 7–9, pp. 730–742, 2006.

[15] F. Ferraty and P. Vieu, "Curves discrimination: a nonparametric functional approach," *Computational Statistics and Data Analysis*, vol. 44, no. 1-2, pp. 161–173, 2003.

[16] J.Q. Shi, R. Murray-Smith, and M. Titterington, "Bayesian regression and classification using mixtures of gaussian process," *International Journal of Adaptive Control and Signal Processing*, vol. 17, pp. 149–161, 2003.