

BAYESIAN UNSUPERVISED SIGNAL CLASSIFICATION BY DIRICHLET PROCESS MIXTURES OF GAUSSIAN PROCESSES

Edmund Jackson¹, Manuel Davy², Arnaud Doucet³, William J. Fitzgerald¹

¹ Signal Processing Group, Univ. of Cambridge Engineering Department, Cambridge, UK

² LAGIS/CNRS/INRIA-FUTURS Sequel, Cité Scientifique, BP 48, 59651 Villeneuve d'Ascq, France

³Dept of Computer Science & Dept of Statistics, Univ. of British Columbia, Vancouver, Canada

ABSTRACT

This paper presents a Bayesian technique aimed at classifying signals without prior training (clustering). The approach consists of modelling the observed signals, known only through a finite set of samples corrupted by noise, as Gaussian processes. As in many other Bayesian clustering approaches, the clusters are defined thanks to a mixture model. In order to estimate the number of clusters, we assume a priori a countably infinite number of clusters, thanks to a Dirichlet process model over the Gaussian processes parameters. Computations are performed thanks to a dedicated Monte Carlo Markov Chain algorithm, and results involving real signals (mRNA expression profiles) are presented.

Index Terms— Clustering, Gaussian Process, Dirichlet Process, MCMC, interpolation.

1. INTRODUCTION

In this paper, we consider the problem of classifying a set of N signals into an unknown number k of classes without prior training (in the following, this problem is termed *clustering*). We adopt a *functional* viewpoint [1] in which a sampled signal is considered as the observed counterpart of an underlying unobserved function. An important point is that the approach can be implemented with signals in one or more dimensions, which may have been sampled at different points. More precisely, consider the set of functions $\mathbf{X} = \{\mathbf{x}_1(\cdot), \dots, \mathbf{x}_N(\cdot)\}$ where each $\mathbf{x}_i(\cdot)$ is a function to be assigned to a class. Each $\mathbf{x}_i(\cdot)$ is known through a set of observed points $(y_{i,j}, t_{i,j})$, $j = 1, \dots, T_i$ called a signal \mathbf{s}_i ($i = 1, \dots, N$) where $y_{i,j} \in \mathbb{Y}$ and $t_{i,j} \in \mathbb{T}$. The space \mathbb{Y} where the $y_{i,j}$ lie is typically \mathbb{R} , while the coordinate space \mathbb{T} is typically \mathbb{R}^d (in the general case, it is assumed to be a Hilbert space). A signal \mathbf{s}_i is related to its underlying function by the relation $y_{i,j} = \mathbf{x}_i(t_{i,j}) + \epsilon_i(t_{i,j})$ where $\epsilon_i(\cdot)$ is a white noise process on \mathbb{T} , assumed Gaussian and zero-mean, with variance σ_ϵ^2 . Moreover, the ϵ_i 's are assumed independent from each other. The problem consists of assigning a class label $z_i = l$ to each $\mathbf{x}_i(\cdot)$ (that is, to each \mathbf{s}_i), $i = 1, \dots, T$, where $l \in \{1, 2, \dots, k\}$. Here, we do not assume that the $t_{i,j}$'s are the same from one signal to another, and we do not need $T_i = T$ for all i .

Functional clustering problems arise in many areas. Example signals are daily load curves of Internet servers, mRNA expression profiles (see Section 5), image textures, etc. Most Bayesian clustering methods consist of defining a mixture model over the data, or over the parameters of some signal model. More precisely, assume the signals follow $\mathbf{s}_i \sim f(\mathbf{s}_i|\theta_i)$ for $i = 1, \dots, N$, where $\theta_i \in \Theta$ is the parameter of some model for the signal \mathbf{s}_i , $f(\cdot|\cdot)$ is the likelihood and \sim means 'distributed according to'. A mixture model is given

by the joint probability distribution function (pdf)

$$\begin{aligned} p(\mathbf{s}_1, \dots, \mathbf{s}_N | k, \omega_{1:k}, \phi_{1:k}) &= \prod_{i=1}^N p(\mathbf{s}_i | k, \omega_{1:k}, \phi_{1:k}) \\ &= \prod_{i=1}^N \sum_{l=1}^k \omega_l f(\mathbf{s}_i | \theta_i = \phi_l) \end{aligned} \quad (1)$$

and it is fully determined by k , $\phi = \{\phi_1, \dots, \phi_k\}$ and $\omega = \{\omega_1, \dots, \omega_k\}$ with $\omega_1 + \dots + \omega_k = 1$. Clustering is performed by noting that writing $\mathbf{s}_i \sim p(\mathbf{s}_i | k, \omega_{1:k}, \phi_{1:k})$ is equivalent to

$$z_i \sim \Pr(z_i | \omega) \quad \text{where} \quad \Pr(z_i = l | \omega) = \omega_l \quad (2)$$

$$\mathbf{s}_i \sim f(\mathbf{s}_i | \theta_i = \phi_{z_i}) \quad (3)$$

and performing inference on the z_i 's. Several approaches have been proposed to tackle the numerical estimation, such as the Expectation-Maximization algorithm. This requires, however, that the number k of clusters is known. Here, we want to avoid this assumption, and we resort to a Bayesian approach, where k is estimated. This can be done by defining a prior distribution over k , and estimating it (together with the z_i 's) by Markov Chain Monte Carlo (MCMC) computations – see [2] for the case where $f(\cdot|\cdot)$ is a one-dimensional Gaussian density. Here, we consider the so-called Dirichlet Process Mixture (DPM) model (see [3] and references therein for an overview), which extends the finite mixture model of Eq. (1), see Subsection 1.1 to an infinite mixture. Also, we note that the mixture model in Eq. (1) applies to our functional clustering if we are able to define the likelihood $f(\cdot|\cdot)$ in a convenient way. This is done via Gaussian Processes (GPs), which are briefly presented in Subsection 1.2 below (see also [4] for an introduction).

1.1. Dirichlet Process Mixtures

Let $(\Theta, \mathcal{B}(\theta))$ be a measurable space, and $G(d\theta) = \sum_{l=1}^k \omega_l \delta_{\phi_l}(d\theta)$ be the *mixing distribution*, where δ is the Dirac delta function. Then in Eq. (1)

$$p(\mathbf{s}_i | k, \omega_{1:k}, \phi_{1:k}) = \sum_{l=1}^k \omega_l f(\mathbf{s}_i | \theta_i = \phi_l) \quad (4)$$

$$= \int_{\Theta} f(\mathbf{s}_i | \theta) G(d\theta) \quad (5)$$

DPMs are defined by replacing $G(d\theta)$ in Eq (5) by the infinite sum

$$G(d\theta) = \sum_{l=1}^{\infty} \omega_l \delta_{\phi_l}(d\theta) \quad (6)$$

where the cluster locations θ_l are distributed according to $G_0(d\theta)$ and the weights ω_l follow the so-called *stick-breaking* representation

$$\omega_l = \beta_l \prod_{m=1}^{l-1} (1 - \beta_m) \text{ and } \beta_l \sim \mathcal{B}(1, \alpha) \quad (7)$$

where \mathcal{B} denotes the Beta distribution. The distribution G defined above (or, equivalently, the set $\{\omega_l, \phi_l\}_{l=1,2,\dots}$) is the random outcome of a so-called Dirichlet Process denoted $\mathcal{DP}(G; \alpha, G_0)$. The resulting infinite mixture model is a DPM with the hierarchical structure:

$$G \sim \mathcal{DP}(G; \alpha, G_0) \quad (8)$$

$$\theta \sim G(d\theta) \quad (9)$$

$$\mathbf{s} \sim f(\mathbf{s}|\theta) \quad (10)$$

and we seen that the class label is implicitly selected in Eq. (9) when θ is sampled. Several signals \mathbf{s}_i 's are generated by iterating Eq.'s (9)–(10), with fixed G . In order to introduce the class labels z , one notes that Eq. (9) may be replaced by

$$z \sim \Pr(z|G) \text{ where } \Pr(z = l|G) = \omega_l \text{ in Eq. (7)} \quad (11)$$

$$\theta = \phi_z \quad (12)$$

When sampling a finite set of signals $\{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ using the DPM model, a finite number k of different z_i 's is used. For large N , $\mathbb{E}[k|\alpha, N] \simeq \alpha \log(1 + \frac{N}{\alpha})$, thus the effective number of classes is tuned by α .

An appealing feature of the DPM model is the so-called Polya urn, which provides the conditional distribution of the parameter θ_i related to a signal \mathbf{s}_i , as follows:

$$p(d\theta|\theta_1, \dots, \theta_N) = \frac{1}{\alpha + N} \left[\sum_{i=1}^N \delta_{\theta_i}(d\theta) + \alpha G_0(d\theta) \right] \quad (13)$$

As can be seen, the Polya urn practically integrates out the Dirichlet Process G and it makes explicit the role of α .

1.2. Gaussian Processes

Gaussian processes [4] extend multivariate Gaussians to (non countably) infinite dimensional random variables. A Gaussian process, denoted $\mathcal{GP}(\mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot))$ is characterized by its mean function $\mathbf{m}(\cdot)$ and its covariance function $\mathbf{K}(\cdot, \cdot)$, and its outcomes are functions. Assume $\mathbf{x}(\cdot)$ is a realisation of a Gaussian process

$$\mathbf{x}(\cdot) \sim \mathcal{GP}(\mathbf{x}(\cdot); \mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot)) \quad (14)$$

then $\{y_j = \mathbf{x}(t_j) + \epsilon(t_j)\}_{j=1,\dots,T}$ is a T -dimensional Gaussian random variable with mean vector $\bar{\mathbf{m}} = \{\mathbf{m}(t_j)\}_{j=1,\dots,T}$ and covariance matrix $\bar{\mathbf{K}} = \{\mathbf{K}(t_j, t_{j'}) + \delta_{t_j, t_{j'}} \sigma_\epsilon^2\}_{(j,j')=1,\dots,T}$. Gaussian processes are useful to interpolate a signal $\mathbf{s}_i = \{(y_{i,j}, t_{i,j}), j = 1, \dots, T_i\}$ at point $t \neq t_{i,j}$, since $p(y|\mathbf{s}_i, t)$ equals

$$\mathcal{N}(y; \mathbf{m}(t) + \bar{\kappa}^T \bar{\mathbf{K}}^{-1}(\bar{\mathbf{y}}_i - \bar{\mathbf{m}}), \mathbf{K}(t, t) - \bar{\kappa}^T \bar{\mathbf{K}}^{-1} \bar{\kappa}) \quad (15)$$

where $\bar{\kappa} = \{\mathbf{K}(t, t_j)\}_{j=1,\dots,T}$, $\bar{\mathbf{y}}_i = \{y_{i,1}, \dots, y_{i,T_i}\}$ and $\mathcal{N}(\cdot; a, b)$ is the Gaussian distribution with mean vector a and covariance matrix b . We see that considering noisy observations is essentially equivalent to considering unnoisy observations, but a slight modification of the covariance function $\mathbf{K}_{\text{noisy}}(t, t') = \mathbf{K}_{\text{unnoisy}}(t, t') + \delta(t = t') \sigma_\epsilon^2$. Thus, in the following, we assume that the signals generated by a GP are unnoisy, without loss of generality. More generally, the covariance function tunes the “roughness” of the realizations $\mathbf{x}(\cdot)$. Almost surely, smooth covariances result in smooth functions while rough covariances give rise to rough functions.

1.3. Contributions and paper organisation

The main contribution of this paper is to propose a model where signals are modelled as GPs whose parameters $\mathbf{m}(\cdot)$ and $\mathbf{K}(\cdot, \cdot)$ determine the clusters. Clustering is performed by applying a DPM over the GP parameters $\mathbf{m}(\cdot)$ and $\mathbf{K}(\cdot, \cdot)$. The complete Bayesian model, referred to as Dirichlet Process Mixture of Gaussian Processes (DPMGP), is derived in Section 2 via the selection of convenient priors. Thanks to an efficient MCMC procedure (Section 3), this model enables the estimation of class labels. It also provides the mean and covariance functions of each class, thus enabling class-dependent signal interpolation. In Section 4, we discuss several features of the model and of the algorithm, and link them to previous works. Simulation results are presented in Section 5, for synthetic signals as well as biological sequences (mRNA expression data). Section 6 presents some conclusions and future work directions.

2. A BAYESIAN DPMGP MODEL

The Bayesian unsupervised classification model is given by

$$G \sim \mathcal{DP}(G; G_0, \alpha) \quad (16)$$

$$\mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot) \sim G(d(\mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot))) \quad (17)$$

$$\mathbf{x}(\cdot) \sim \mathcal{GP}(\mathbf{x}(\cdot), \mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot)) \quad (18)$$

$$y_j = \mathbf{x}(t_j) \text{ for } j = 1, \dots, T \quad (19)$$

An equivalent model can be written by letting the cluster variable z appear in Eq. (17), see Subsection 1.1. In order to fully define this model, we need to select the prior $G_0(\mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot))$. Here, we select the following hierarchical shape

$$G_0(\mathbf{m}(\cdot), \mathbf{K}(\cdot, \cdot)) = G_0^1(\mathbf{m}(\cdot)|\mathbf{K}(\cdot, \cdot)) G_0^2(\mathbf{K}(\cdot, \cdot)) \quad (20)$$

2.1. Prior distribution for the mean function

A very natural choice for $G_0^1(\mathbf{m}(\cdot)|\mathbf{K}(\cdot, \cdot))$ is the zero-mean GP with covariance function $\mathbf{K}(\cdot, \cdot)$

$$G_0^1(\mathbf{m}(\cdot)|\mathbf{K}(\cdot, \cdot)) = \mathcal{GP}(\mathbf{m}(\cdot); 0, \mathbf{K}(\cdot, \cdot)) \quad (21)$$

Another possibility is to model the mean function by a parametric model. For the simulations presented in Section 5, we have chosen to select the most simple model by letting $G_0^1(\mathbf{m}(\cdot)|\mathbf{K}(\cdot, \cdot)) = \delta_0(\mathbf{m}(\cdot))$ (that is, by considering that the functions \mathbf{x}_i 's are generated from zero-mean GPs). Indeed, in the applications presented, simulations showed that the covariance function is a much more discriminant feature than the mean, thus $\mathbf{m}(\cdot)$ has less importance, and it can be forced to zero without lowering the performance.

2.2. Prior distribution for the covariance function

Following several previous works [5], we adopt the following parametric shape for the covariance function (assuming $\mathbb{T} = \mathbb{R}^d$ for presentation simplicity)

$$\mathbf{K}(t_1, t_2) = a_0 + a_1 \sum_{q=1}^d t_1^q t_2^q + a_2 \exp \left(-\frac{1}{2} \sum_{q=1}^d b_q |t_1^q - t_2^q|^2 \right) \quad (22)$$

where t_j^q is component $\#q$ in vector t_j . Eq. (22) is a linear combination of the linear covariance and the Gaussian covariance, thus ensuring that $\mathbf{K}(\cdot, \cdot)$ is indeed definite-positive [6]. This model can be easily generalized to linear combinations of other kind of covariance (for example, a periodic covariance function is used in Subsection 5.2). Also, a diagonal term $\sigma_\epsilon^2 \delta_{t_1, t_2}$ could be added to take account of the observation noise ϵ . In order to fully define $G_0^2(\mathbf{K}(\cdot, \cdot))$,

we need to assign prior distributions to the parameter $\theta = \{a_0, a_1, a_2, b_1, \dots, b_d\}$. We assign an inverse Gamma prior over each of the components of θ , thus $G_0^2(\mathbf{K}(\cdot, \cdot)) = G_0^2(\theta)$ is the product of $d + 3$ inverse Gamma distributions.

2.3. Estimation objectives

Given parametric models for $\mathbf{m}(\cdot)$ and $\mathbf{K}(\cdot, \cdot)$, and N signals $\mathbf{s}_1, \dots, \mathbf{s}_N$, the objective is to estimate the N class labels $\mathbf{z} = \{z_i, i = 1, \dots, N\}$ as well as the locations $\phi = \{\phi_l, l = 1, \dots, k\}$ such that $\theta_i = \phi_{z_i}$. The z_i 's provide the required classification while the ϕ_l 's give the GP parameters that best explain the signals such that $z_i = l$. In the following, we denote $\mathbf{m}_l(\cdot)$ and $\mathbf{K}_l(\cdot, \cdot)$ the mean and covariance functions computed with the parameters $\theta = \phi_l$.

3. A MCMC ALGORITHM FOR DPMGP

In this section, we describe the MCMC algorithm used to generate samples $\tilde{z}_i^{(n)}, \tilde{\phi}_i^{(n)}$ ($n = 1, 2, \dots$) from their joint posterior probability, to be used to estimate \mathbf{z} and ϕ from the signals \mathbf{s}_i 's. Following [7, Algorithm 5]), we implement a Gibbs sampler which iteratively samples from the conditional probability $\Pr(\mathbf{z}|\phi, \mathbf{s}_1, \dots, \mathbf{s}_N)$ and the conditional pdf $p(\phi|\mathbf{z}, \mathbf{s}_1, \dots, \mathbf{s}_N)$.

Let $\mathcal{I}(\mathbf{z})$ denote the set of values taken by the variables z_1, \dots, z_L . Indeed, for the sake of presentation simplicity, we no longer assume that the z_i 's take all the values in a set $\{1, \dots, k\}$, but instead some values spread among the integers. The reason for this is that the Gibbs samplers may create and suppress locations ϕ_l 's, thus incrementing the largest l while forming gaps in-between indexes l . Thus, $\mathcal{I}(\mathbf{z})$ contains the values of l that are effectively used at a given iteration. We denote $N_{-i,l}(\mathbf{z}) = \sum_{i'=1, i' \neq i}^N \delta_{l, z_{i'}}$ the number of $z_{i'}$'s ($i' \neq i$) which equal l . The likelihood is denoted by $f(\mathbf{s}|\phi_z)$ with $f(\mathbf{s}|\phi_z) = \mathcal{N}(\mathbf{s}; \bar{\mathbf{m}}_z, \bar{\mathbf{K}}_z)$ where the mean vector is $\bar{\mathbf{m}}_z = \{\mathbf{m}_z(t_1), \dots, \mathbf{m}_z(t_T)\}$ and the covariance matrix is $\bar{\mathbf{K}}_z = \{\mathbf{K}_z(t_j, t_{j'}), (j, j') = 1, \dots, T\}$.

In Algorithm 1 below, the signals are first assigned to some class, then class parameter location are updated conditional on the related signals. In practice, this sampling scheme is quite efficient, and convergence is reached quickly. The proposal density $q(\phi^*|\phi'_l)$ may be randomly selected as a Gaussian random walk, i.e. $q(\phi^*|\phi'_l) = \mathcal{N}(\phi^*; \phi'_l, \Sigma_{\text{rw}})$ or as the prior distribution, i.e. $q(\phi^*|\phi'_l) = G_0(\phi^*)$. Further details about the algorithm may be found in [7].

Algorithm 1: Gibbs sampler for the DPMGP model

Step 1: Initialization

- For $i = 1, \dots, N$, sample $\tilde{\theta}_i^{(0)} \sim p(\theta_i|\theta_1, \dots, \theta_{i-1})$ given in Eq.(13) and deduce $\tilde{\mathbf{z}}^{(0)}$ and $\tilde{\phi}^{(0)}$.

Step 2: iterations. For $n = 1, 2, \dots$, do

% Step 2.1: Sample from $\Pr(\mathbf{z}|\tilde{\phi}^{(n-1)}, \mathbf{s}_1, \dots, \mathbf{s}_N)$ as follows

- Let $\mathbf{z}' \leftarrow \tilde{\mathbf{z}}^{(n-1)}$ and let $\phi' \leftarrow \tilde{\phi}^{(n-1)}$
- For $i = 1, \dots, N$, update z'_i by the following MH step
 - Sample a candidate z_i^* from the Polya Urn probabilities:

$$Q(z_i^* = l) = \begin{cases} \frac{N_{-i,l}(\mathbf{z}')}{\alpha + N - 1} & \text{for } l \in \mathcal{I}(\mathbf{z}') \\ \frac{\alpha}{\alpha + N - 1} & \text{for a new } l \notin \mathcal{I}(\mathbf{z}') \\ 0 & \text{for all other values of } l \end{cases} \quad (23)$$

- if $z_i^* \in \mathcal{I}(\mathbf{z}')$, then compute $r_1 = f(\mathbf{s}_i|\phi'_{z_i^*})/f(\mathbf{s}_i|\phi'_{z'_i})$; otherwise, compute $r_1 = f(\mathbf{s}_i|\phi^*)/f(\mathbf{s}_i|\phi'_{z'_i})$ where $\phi^* \sim G_0$. With probability $\min(1, r_1)$, set $z'_i \leftarrow z_i^*$ and $\phi'_{z'_i} \leftarrow \phi^*$.

- Let $\tilde{z}_i^{(n)} \leftarrow z'_i$

% Step 2.2: For $l \in \mathcal{I}(\tilde{\mathbf{z}}^{(n)})$, sample $\tilde{\phi}_l^{(n)}$ from $p(\phi_l|\mathbf{s}_i$ with i such that $\tilde{z}_i^{(n)} = l$), as follows

- Sample $\phi^* \sim q(\phi^*|\phi'_l)$
- Compute

$$r_2 = \frac{q(\phi'_l|\phi^*)}{q(\phi^*|\phi'_l)} \frac{G_0(\phi^*)}{G_0(\phi'_l)} \prod_{\substack{i=1, \dots, N \\ \text{such that } \tilde{z}_i^{(n)} = l}} \frac{f(\mathbf{s}_i|\phi^*)}{f(\mathbf{s}_i|\phi'_l)} \quad (24)$$

and, with probability $\min(1, r_2)$, set $\tilde{\phi}_l^{(n)} \leftarrow \phi^*$; otherwise, set $\tilde{\phi}_l^{(n)} \leftarrow \phi'_l$

4. DISCUSSION

The model presented in this paper has several practical advantages. Firstly, it can be implemented on signals in one or more dimensions without much algorithm changes. Second, the GP, together with a given G_0 , enables the smoothness/roughness properties to be discriminant or not. After assignment of a signal to a given class, the GP enables signal interpolation with class-specific parameters, which may be more robust than GP regression with blindly selected parameters. Also, the signals \mathbf{s}_i to be classified do not need to have been sampled at the same points $t_{i,j}$ ($j = 1, \dots, T_i$), with possibly different lengths T_i 's. This latter property is of great interest in many applications, where one collects irregularly sampled signals and tries to class them into groups.

Aside Algorithm 1 above, we have implemented and tested the retrospective sampling approach of [8]. This latter algorithm is more complex and provides results similar to Algorithm 1. A common difficulty in Bayesian clustering is the so-called *label-switching problem* [9]: from one iteration of Algorithm 1 to another, the class labels and parameter locations may change, and equivalent labellings may be numerically different (because of shifts and permutations). This requires the implementation of an additional layer to post-process the MCMC samples. A solution to overcome this problem is to embed the Gibbs sampler in Algorithm 1 into a simulated annealing procedure, or take one sample after convergence of the Markov Chain (which is what we did in simulations).

Several previous works are related to this approach. The most closely related is that of Shi et al. [5], which investigates regression with a finite mixture of Gaussian Processes.

5. RESULTS

In this section, we first propose results obtained with synthetic data. Then, we provide results obtained from mRNA signals.

5.1. Synthetic signals

To demonstrate the effectiveness of the proposed approach, we have produced a set comprising of 40 one-dimensional signals from four different GPs (10 from each class) with zero-mean and covariance function given in Eq.(22), where $a_0 = a_1 = 0$.

Class tag	1	2	3	4
True a_2	0.8	0.6	0.4	0.2
True b_1	0.01	0.04	0.08	0.12
Estimated a_2	0.97	0.79	0.396	0.19
Estimated b_1	0.01	0.04	0.08	0.106

Table 1. Covariance function parameters used to generate the synthetic data, and estimated class locations.

The priors over a_2 and b_1 are the inverse-gamma distributions with parameters $\alpha = 0.65$, $\beta = 20$, which provides support over the realised values. This dataset has been processed with Algorithm 1 with $\alpha = 0.25$, resulting in the following classification results. Six classes have been identified with locations. Classes #1, #2 and #3 contain 100% well classified signals, whereas class #4 contains 80% of the signals simulated with its parameters. The two remaining signals are assigned to individual classes with parameters ($a_2 = 0.11$, $b_1 = 0.10$) and ($a_2 = 0.08$, $b_1 = 0.07$).

5.2. mRNA expression profile signals

A problem of interest in systems- and cell-biology is that of identifying cell-cycle regulated genes, which is still an open problem [10]. The problem amounts to identifying those genes which exhibit a periodic gene expression time-series. Traditional approaches to this problem employ Fourier analysis, spline regression and other similar ideas. Here we propose to cluster the observed expression function using DPMGPs based on the covariance which incorporates a periodic term to model the regulated genes, and an aperiodic term to model the unregulated genes

$$\mathbf{K}(t_1, t_2) = a_2 \exp\left(-\frac{b}{2} \sum_{q=1}^d (t_1^q - t_2^q)^2\right) + a_3 \exp\left(-\frac{1}{2} \sum_{q=1}^d \sin 2\pi c(t_1^q - t_2^q)^2\right), \quad (25)$$

We extracted 200 mRNA expression profiles from the *Saccharomyces cerevisiae* cdc28 arrest experiment [10], each comprising 17 time-points. The algorithm was run with $\alpha = 0.25$ for 2000 iterations, $b \triangleq 0.03$ and an inverse gamma prior with parameters for a_2 , a_3 and c : (10, 2), (4, 0.4) and (5, 0.7) respectively. Figure 1 illustrates two representative clusters. Although performance is impossible to quantify as no ground truth parameter values are available, the discriminatory ability of the algorithm is apparent through the separation of the periodic from the aperiodic functions.

6. CONCLUSION

The original Bayesian unsupervised classification method presented in this paper is shown to be efficient in front of synthetic and real signals. Further investigations will consider larger classes of parametric covariance functions, as well as image-related clustering problems.

7. REFERENCES

- [1] J. Ramsay and BW Silverman, *Functional Data Analysis*, Springer, 2005.
- [2] S. Richardson and P. J Green, "On Bayesian analysis of mixtures with an unknown number of components," *J. Royal Stat. Soc. B*, vol. 59, no. 4, pp. 731–792, 1997.
- [3] M.D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *J. Am. stat. assoc.*, vol. 90, pp. 577–588, 1995.
- [4] D.J.C. MacKay, "Introduction to Gaussian processes," *Neural Networks and Machine Learning*, vol. 168, 1998.
- [5] J.Q. Shi, R. Murray-Smith, and D.M. Titterton, "Hierarchical Gaussian process mixtures for regression," *Stat. Comp.*, vol. 15, pp. 31–41, 2005.
- [6] N. Aronszajn, "Theory of reproducing kernels," *Trans. Am. Math. Soc.*, vol. 68, pp. 337–404, 1950.
- [7] R.M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," Tech. Rep. 9815, Dpt Stat. and dpt comp. sc., Univ. Toronto, Canada, 1998.
- [8] O. Papaspiliopoulos and G.O. Roberts, "Retrospective Markov chain Monte Carlo method for Dirichlet process hierarchical priors," *Biometrika*, 2005, Submitted.
- [9] M. Stephens, "Dealing with label switching in mixture models," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 62, no. 4, pp. 795–809, 2000.
- [10] U. de Lichtenberg, L. Jensen, Fausboll A., Jensen T.S., Bork P., and Brunak S., "Comparision of computational methods for the identification of cell cycle-regulated genes," *Bioinformatics*, vol. 21, no. 7, pp. 1164–1171, 2005.

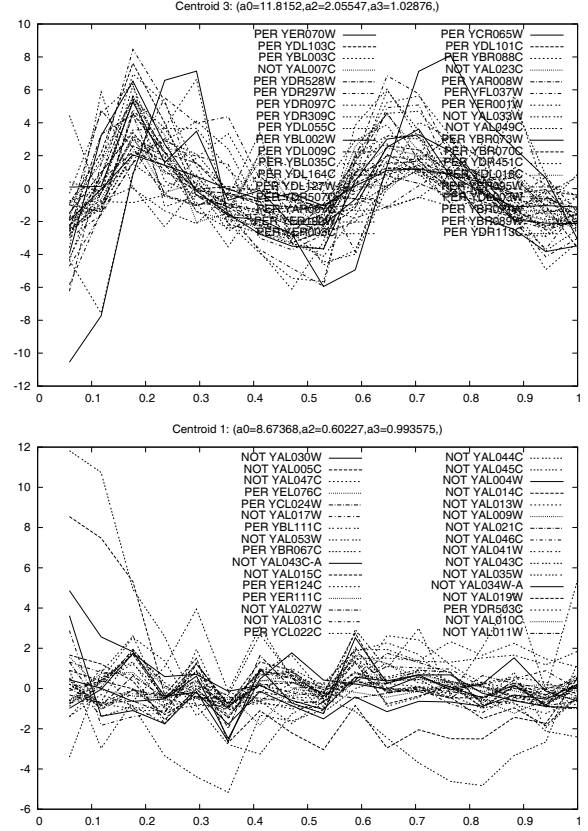


Fig. 1. Gene Expression Data, Classification Results