Kernel discriminant analysis and clustering with parsimonious Gaussian process models

Charles Bouveyron, Mathieu Fauvel, Stéphane Girard

Kernel discriminant analysis and clustering with parsimonious Gaussian process models

C. Bouveyron[1], M. Fauvel[2] & S. Girard[3]

[1] Laboratoire MAP5, UMR 8145, Université Paris Descartes & Sorbonne Paris Cité
[2] Laboratoire DYNAFOR, UMR 1201, INRA & Université de Toulouse
[3] Equipe MISTIS, INRIA Grenoble Rhône-Alpes & LJK
FRANCE

This work presents a family of parsimonious Gaussian process models which allow to build, from a finite sample, a model-based classifier in an infinite dimensional space. The proposed parsimonious models are obtained by constraining the eigen-decomposition of the Gaussian processes modeling each class. This allows in particular to use non-linear mapping functions which project the observations into infinite dimensional spaces. It is also demonstrated that the building of the classifier can be directly done from the observation space through a kernel function. The proposed classification method is thus able to classify data of various types such as categorical data, functional data or networks. Furthermore, it is possible to classify mixed data by combining different kernels. The methodology is as well extended to the unsupervised classification case and an EM algorithm is derived for the inference. Experimental results on various data sets demonstrate the effectiveness of the proposed method. *A Matlab toolbox implementing the proposed classification methods is provided as supplementary material.*

1 Introduction

Classification is an important and useful statistical tool in all scientific fields where decisions have to be made. Depending on the availability of a learning data set, two main situations may happen: supervised classification (also known as discriminant analysis) and unsupervised classification (also known as clustering). Discriminant analysis aims to build a classifier (or a decision rule) able to assign an observation x in an arbitrary space E with unknown class membership to one of k known classes $C_1, ..., C_k$. For building this supervised classifier, a learning dataset $\{(x_1, z_1), ..., (x_n, z_n)\}$ is used, where the observation $x_\ell \in E$ and $z_\ell \in \{1, ..., k\}$ indicates the class belonging of the observation x_ℓ. In a slightly different context, clustering aims to directly partition an incomplete dataset $\{x_1, ..., x_n\}$ into k homogeneous groups without any other information, *i.e.,* assign to each observation $x_\ell \in E$ its group membership $z_\ell \in \{1, ..., k\}$. Several intermediate situations exist, such as semi-supervised or weakly-supervised classifications (Chapelle et al., 2006), but they will not be considered here.

Since the pioneer work of Fisher (Fisher, 1936), a huge number of supervised and unsupervised classification methods have been proposed in order to deal with different types of data. Indeed, there exist a wide variety of data such as quantitative, categorical and binary data but also texts, functions, sequences, images and more recently networks. As a practical example, biologists are frequently interested

in classifying biological sequences (DNA sequences, protein sequences), natural language expressions (abstracts, gene mentioning), networks (gene interactions, gene co-expression), images (cell imaging, tissue classification) or structured data (gene structures, patient information). The observation space E can be therefore $\mathbb{R}^p$ if quantitative data are considered, $L^2([0,1])$ if functional data are considered (time series for example) or $\mathcal{A}^p$, where $\mathcal{A}$ is a finite alphabet, if the data at hand are categorical (DNA sequences for example). Furthermore, the data to classify can be a mixture of different data types: categorical and quantitative data or categorical and network data for instance.

Classification methods can be split into two main families: generative and discriminative techniques. On the one hand, generative or model-based techniques model the data of each class with a probability distribution and deduce the classification rule from this modeling. On the other hand, discriminative techniques directly build the classification rule from the learning dataset. Among the discriminative methods, kernel methods and Gaussian process classification are the most popular. Section 2 briefly reviews these techniques.

In this work, we propose to adapt model-based methods for the classification of any kind of data by working in a feature space of high or even infinite dimensional space. To this end, we propose a family of parsimonious Gaussian process models which allow to build, from a finite sample, a model-based classifier in a infinite dimensional space. It will be demonstrated that the building of the classifier can be directly done from the observation space through the so-called "kernel trick". The proposed classification method will be thus able to classify data of various types (categorical data, mixed data, functional data, networks, etc). The methodology is as well extended to the unsupervised classification case (clustering).

The paper is organized as follows. Section 2 reviews generative and discriminative techniques for classification. Section 3 then presents the context of our study and introduces the family of parsimonious Gaussian process models. The inference aspects are addressed in Section 4. It is also demonstrated in this section that the proposed method can work directly from the observation space through a kernel. Section 5 is dedicated to some special cases and to the extension to the unsupervised framework through the derivation of an EM algorithm. *Experimental comparisons with state-of-the-art kernel methods are presented in Section 6 as well as applications of the proposed methodologies to various types of data including functional, categorical and mixed data.* Some concluding remarks are given in Section 7 and proofs are postponed to the appendix.

2 Related work

This section briefly reviews model-based and discriminative techniques for classification.

2.1 Model-based classification

Model-based discriminant analysis assumes that $\{x_1, ..., x_n\}$ are independent realizations of a random vector X on E and that the class conditional distribution of X is parametric, *i.e.* $f(x|z=i) = f_i(x;\theta_i)$. When $E = \mathbb{R}^p$, among the possible parametric distributions for f_i, the Gaussian distribution is often preferred and, in this case, the marginal distribution of X is therefore a mixture of Gaussian distributions:

$$f(x) = \sum_{i=1}^{k} \pi_i \phi(x;\mu_i,\Sigma_i),$$

where ϕ is the Gaussian density, π_i is the prior probability of the ith class, μ_i is the mean of the ith class and Σ_i is its covariance matrix. The optimal decision rule is called the *maximum a posteriori* (MAP) rule which assigns a new observation x to the class with the largest posterior probability. Introducing the classification function $D_i(x) = \log|\Sigma_i| + (x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i) - 2\log(\pi_i)$, which can be rewritten as:

$$D_i(x) \quad = \quad \sum_{j=1}^{p} \frac{1}{\lambda_{ij}} < x-\mu_i, q_{ij} >^2_{\mathbb{R}^p} + \sum_{j=1}^{p} \log(\lambda_{ij}) - 2\log(\pi_i), \tag{1}$$

where q_{ij} and λ_{ij} are respectively the jth eigenvector and eigenvalue of Σ_i, it can be easily shown that the MAP rule reduces to finding the label $i \in \{1,\ldots,k\}$ for which $D_i(x)$ is the smallest. Estimation of model parameters is usually done by maximum likelihood. This method is known as the quadratic discriminant analysis (QDA), and, under the additional assumption that $\Sigma_i = \Sigma$ for all $i \in \{1,\ldots,k\}$, it corresponds to the linear discriminant analysis (LDA). A detailed overview on this topic can be found in (McLachlan, 1992). Recent extensions allowing to deal with high-dimensional data include (Bouveyron and Brunet, 2012; Bouveyron et al., 2007b,a; McLachlan et al., 2003; McNicholas and Murphy, 2008; Montanari and Viroli, 2010; Murphy et al., 2010) and an extensive review of those techniques is given in (Bouveyron and Brunet-Saumard, 2013). Although model-based classification is usually enjoyed for its multiple advantages, model-based discriminant analysis methods is mostly limited to quantitative data. Some extensions exist to handle categorical data using multinomial (Celeux and Govaert, 1991) or Dirichlet (Bouguila et al., 2003) distributions for instance. In addition, even in the case of quantitative data, the Gaussian assumption may not be well-suited for the data at hand. Recent works focused on different distributions such as the skew normal (Lin et al., 2007), asymmetric Laplace (Franczak et al., 2014) or t-distributions (Andrews and McNicholas, 2012; Lee and McLachlan, 2013; Lin, 2010; Forbes and Wraith, 2014).

2.2 Kernel methods for classification

Among discriminative classification methods, kernel methods (Hofmann et al., 2008) are probably the most popular and overcome some of the shortcomings of generative techniques. Kernel methods are non-parametric techniques and can be applied to any data for which a kernel function can be defined. A kernel $K : E \times E \to \mathbb{R}$ is a positive definite function such as every evaluation can be written as $K(x_i, x_j) =< \varphi(x_i), \varphi(x_j) >_{\mathcal{H}}$, with $x_i, x_j \in E$, φ a mapping function (called the feature map), $\mathcal{H}$ a finite or infinite dimensional reproducing kernel Hilbert space (the feature space) and $< \cdot, \cdot >_{\mathcal{H}}$ the dot product in $\mathcal{H}$. An advantage of using kernels is the possibility of computing the dot product in the feature space from the original input space without explicitly knowing φ (kernel trick) (Hofmann et al., 2008). Turning conventional learning algorithms into kernel learning algorithms can be done if the algorithms operate on the data only in terms of dot product. For instance, kernels can be defined on strings (Shawe-Taylor and Cristianini, 2004; Cuturi and Vert, 2005), graphs (Smola and Kondor, 2003; Mahé and Vert, 2009), trees (Schölkopf et al., 2004, Chap. 5) and vector-valued functions (Caponnetto et al., 2008; Evgeniou et al., 2005; Kadri et al., 2012). Many conventional linear algorithms have been turned to non-linear algorithms thanks to kernels (Scholkopf and Smola, 2001): KPCA (Schölkopf et al., 1998), KFD (Mika et al., 1999), KGMM (Dundar and Landgrebe, 2004), mixture of Gamma distributions (Murua and Wicker, 2014) and others. Let us however highlight that those kernel versions often involve the inversion of a kernel matrix, *i.e.*, a $n \times n$ matrix estimated with only n samples. Usually, the kernel matrix is ill-conditioned and regularization is needed, while sometimes a simplified

model is required too. Thus, it may limit the effectiveness of the kernel version. In addition, and conversely to model-based techniques, classification results provided by kernel methods are difficult to interpret which would be useful in many application domains.

2.3 Classification with Gaussian processes

A recent approach in discriminative classification is to make use of Gaussian processes (GP) in a Bayesian view of logistic regression. In this context, Gaussian processes are used as prior distribution for the regression function which links the predictive variable (Z) to the descriptive vector (X). The conditional distribution of Z is then given through the relation $\text{logit}\left(p(Z=1|X=x)\right) = f(x)$ where the prior distribution on the regression function f is $f \sim GP(0,\Sigma)$ with Σ a given covariance function. In the regression context, the inference is usually tractable since the Gaussian process prior is combined with a Gaussian likelihood. Unfortunately, in the classification case, the discrete nature of the predictive variable makes the exact inference infeasible. To overcome this difficulty, several approximation techniques have been proposed lately, such as the Laplace approximation (Kuss and Rasmussen, 2005) or through the expectation-propagation algorithm (Minka, 2001). Let us notice that the latter approximation is usually more accurate in term of classification performance than the former in practical situations (Rasmussen and Williams, 2006b). Furthermore, a limiting point of Gaussian process classification is, once again, the need to numerically invert a $n \times n$ covariance matrix. Thus, the inference of a Gaussian process predictor, even in a feasible case, is computationally demanding ($O(n^3)$ where n is the number of learning observations). We refer to (Rasmussen and Williams, 2006b, Chapter 3) for an extensive overview of Bayesian classification with Gaussian processes.

3 Classification with parsimonious Gaussian process models

In this section, we first define the context of our approach and exhibit the associated computational problems. Then, a parsimonious parameterization of Gaussian processes is proposed in order to overcome the highlighted computational issues. Before to move forward and in order to avoid any misunderstanding, we would like to emphasize that we do not consider in this work Gaussian processes in a Bayesian regression framework. In our approach, the Gaussian processes are used as conditional distributions of a latent process and not as prior distributions for Bayesian logistic regression.

3.1 Classification with Gaussian processes

Let us consider a learning set $\{(x_1,z_1),...,(x_n,z_n)\}$ where $\{x_1,...,x_n\} \subset E$ are assumed to be independent realizations of a, possibly non-quantitative and non-Gaussian, random variable X. The class labels $\{z_1,...,z_n\}$ are assumed to be realizations of a discrete random variable $Z \in \{1,...,k\}$. It indicates the memberships of the learning data to the k classes denoted by $C_1,\ldots,C_k$, *i.e.,* $z_\ell = i$ indicates that x_ℓ belongs to C_i.

Let us assume that there exists a non-linear mapping φ such that the latent process $Y = \varphi(X)$ is, conditionally on $Z=i$, a Gaussian process on $J \subset \mathbb{R}$ with mean μ_i and continuous covariance function Σ_i. Hence, the use of the transformation φ allows to embed a possible non-quantitative vector into a quantitative space, as in kernel methods. More specifically, one has $\mu_i(t) = \mathbb{E}(Y(t)|Z=i)$ and $\Sigma_i(s,t) = \mathbb{E}(Y(s)Y(t)|Z=i) - \mu_i(t)\mu_i(s)$. It is then well-known (Shorack and Wellner, 1986) that, for all $i=1,\ldots,k$, there exist positive eigenvalues (sorted in decreasing order) $\{\lambda_{ij}\}_{j\geq 1}$, together with

eigenvector functions $\{q_{ij}(.)\}_{j\geq 1}$ continuous on J, such that

$$\Sigma_i(s,t) = \sum_{j=1}^{\infty} \lambda_{ij} q_{ij}(s) q_{ij}(t),$$

where the series is uniformly convergent on $J \times J$. Moreover, the eigenvector functions are orthonormal in $L^2(J)$ for the dot product $< f, g >_{L_2} = \int_J f(t)g(t)dt$. It is then easily seen, that, for all $r \geq 1$ and $i \in \{1, \ldots, k\}$, the random vector on $\mathbb{R}^r$ defined by $\{< Y, q_{ij} >_{L_2}\}_{j=1,\ldots,r}$ is, conditionally on $Z = i$, Gaussian with mean $\{< \mu_i, q_{ij} >\}_{j=1,\ldots,r}$ and covariance matrix $\mathrm{diag}(\lambda_{i1}, \ldots, \lambda_{ir})$. To classify a new observation x, we therefore propose to apply the Gaussian classification function (1) to $\varphi(x)$:

$$D_i(\varphi(x)) = \sum_{j=1}^{r} \frac{1}{\lambda_{ij}} < \varphi(x) - \mu_i, q_{ij} >_{L_2}^2 + \sum_{j=1}^{r} \log(\lambda_{ij}) - 2\log(\pi_i).$$

From a theoretical point of view, if the Gaussian process is non degenerated, one should use $r = +\infty$. In practice, r has to be large in order not to loose to much information on the Gaussian process. Unfortunately, in this case, the above quantities cannot be estimated from a finite sample set. Indeed, only a part of the classification function can be actually computed from a finite sample set:

$$\begin{aligned} D_i(\varphi(x)) &= \underbrace{\sum_{j=1}^{r_i} \frac{1}{\lambda_{ij}} < \varphi(x) - \mu_i, q_{ij} >_{L_2}^2 + \sum_{j=1}^{r_i} \log(\lambda_{ij}) - 2\log(\pi_i)}_{\text{computable quantity}} \\ &+ \underbrace{\sum_{j=r_i+1}^{r} \frac{1}{\lambda_{ij}} < \varphi(x) - \mu_i, q_{ij} >_{L_2}^2 + \sum_{j=r_i+1}^{r} \log(\lambda_{ij})}_{\text{non computable quantity}}, \end{aligned}$$

where $r_i = \min(n_i, r)$ and $n_i = \mathrm{Card}\,(C_i)$ is the cardinality of the ith class. Consequently, the Gaussian model cannot be used directly in the feature space to classify data if $r > n_i$ for $i = 1, ..., k$.

3.2 A parsimonious Gaussian process model

To overcome the computation problem highlighted above, it is proposed here to use in the feature space a parsimonious model for the Gaussian process modeling each class. Following the idea of (Bouveyron et al., 2007a), we constrain the eigen-decomposition of the Gaussian processes as follows.

Definition 1. *A parsimonious Gaussian process (pgp) model is a Gaussian process Y for which, conditionally to $Z = i$, the eigen-decomposition of its covariance operator Σ_i is such that:*

(A1) it exists a dimension $r < +\infty$ such that $\lambda_{ij} = 0$ for $j \geq r$ and for all $i = 1, ..., k$,

(A2) it exists a dimension $d_i < \min(r, n_i)$ such that $\lambda_{ij} = \lambda$ for $d_i < j < r$ and for all $i = 1, ..., k$.

It is worth noticing that r can be as large as it is desired, as long it is finite, and in particular r can be much larger than n_i, for any $i = 1, ..., k$. From a practical point of view, this modeling can be viewed as assuming that the data of each class live in a specific subspace of the feature space. The variance of the actual data of the ith group is modeled by the parameters $\lambda_{i1}, ..., \lambda_{id_i}$ and the variance of the noise is modeled by λ. This assumption amounts to supposing that the noise is homoscedastic and its variance is common to all the classes. The dimension d_i can be also considered as the intrinsic dimension of the latent subspace of the ith group in the feature space. This model is referred to as

Model	Variance inside the subspace F_i	Variance outside F_i	Subspace orientation Q_i	Intrinsic dimension d_i
$\mathcal{M}_0$	Free	Common	Free	Free
$\mathcal{M}_1$	Free	Common	Free	Common
$\mathcal{M}_2$	Common within groups	Common	Free	Free
$\mathcal{M}_3$	Common within groups	Common	Free	Common
$\mathcal{M}_4$	Common between groups	Common	Free	Common
$\mathcal{M}_5$	Common within and between groups	Common	Free	Free
$\mathcal{M}_6$	Common within and between groups	Common	Free	Common
$\mathcal{M}_7$	Common between groups	Common	Common	Common
$\mathcal{M}_8$	Common within and between groups	Common	Common	Common

Table 1: Submodels of the parsimonious Gaussian process model (referred to as $\mathcal{M}_0$ here).

pgp$\mathcal{M}_0$ (or $\mathcal{M}_0$ for short) hereafter. With these assumptions, we have the following result *(proof is provided in the appendix)*.

Proposition 1. *Letting $d_{\max} = \max(d_1, ..., d_k)$, the classification function D_i can be written as follows in the case of a parsimonious Gaussian process model pgp$\mathcal{M}_0$:*

$$\begin{aligned} D_i(\varphi(x)) &= \sum_{j=1}^{d_i} \left(\frac{1}{\lambda_{ij}} - \frac{1}{\lambda} \right) < \varphi(x) - \mu_i, q_{ij} >_{L_2}^2 + \frac{1}{\lambda} ||\varphi(x) - \mu_i||_{L_2}^2 \\ &+ \sum_{j=1}^{d_i} \log(\lambda_{ij}) + (d_{\max} - d_i) \log(\lambda) - 2 \log(\pi_i) + \gamma, \end{aligned} \tag{2}$$

where γ is a constant term which does not depend on the index i of the class.

At this point, it is important to notice that the classification function D_i depends only on the eigenvectors associated with the d_i largest eigenvalues of Σ_i. This estimation is now possible due to the inequality $d_i < n_i$ for $i = 1, ..., k$. Furthermore, the computation of the classification function does not depend any more on the parameter r. As shown in Section 4, it is possible to reformulate the classification function such that it does not depend either on the mapping function φ.

3.3 Submodels of the parsimonious model

By fixing some parameters to be common within or between classes, it is possible to obtain particular models which correspond to different regularizations. Table 1 presents the 8 additional models which can be obtained by constraining the parameters of model $\mathcal{M}_0$. For instance, fixing the dimensions d_i to be common between the classes yields the model $\mathcal{M}_1$. Similarly, fixing the first d_i eigenvalues to be common within each class, we obtain the more restricted model $\mathcal{M}_2$. It is also possible to constrain the first d_i eigenvalues to be common between the classes (models $\mathcal{M}_4$ and $\mathcal{M}_7$), and within and between the classes (models $\mathcal{M}_5$, $\mathcal{M}_6$ and $\mathcal{M}_8$). This family of 9 parsimonious models should allow the proposed classification method to fit into various situations.

4 Model inference and classification with a kernel

This section focuses on the inference of the parsimonious models proposed above and on the classification of new observations through a kernel. Model inference is only presented for the model $\mathcal{M}_0$

since inference for the other parsimonious models is similar. Estimation of intrinsic dimensions and visualization in the feature subspaces are also discussed.

4.1 Estimation of model parameters

In the model-based classification context, parameters are usually estimated by their empirical counterparts (McLachlan, 1992) which leads, in the present case, to estimate the proportions π_i by $\hat{\pi}_i = n_i/n$ and the mean function μ_i by $\hat{\mu}_i(t) = \frac{1}{n_i} \sum_{x_j \in C_i} \varphi(x_j)(t)$, where n_i is the number of observations in the ith class. Regarding the covariance operator, the eigenvalue λ_{ij} and the eigenvector q_{ij} are respectively estimated by the jth largest eigenvalue $\hat{\lambda}_{ij}$ and its associated eigenvector function $\hat{q}_{ij}$ of the empirical covariance operator $\hat{\Sigma}_i$:

$$\hat{\Sigma}_i(s,t) = \frac{1}{n_i} \sum_{x_\ell \in C_i} \varphi(x_\ell)(s)\varphi(x_\ell)(t) - \hat{\mu}_i(s)\hat{\mu}_i(t).$$

Finally, the estimator of λ is:

$$\hat{\lambda} = \frac{1}{\sum_{i=1}^{k} \hat{\pi}_i \left(r - d_i\right)} \sum_{i=1}^{k} \hat{\pi}_i \left(\operatorname{trace}(\hat{\Sigma}_i) - \sum_{j=1}^{d_i} \hat{\lambda}_{ij} \right). \tag{3}$$

Using the plug-in method, the estimated classification function $\hat{D}_i$ can be written as follows:

$$\begin{aligned} \hat{D}_i(\varphi(x)) &= \sum_{j=1}^{d_i} \left(\frac{1}{\hat{\lambda}_{ij}} - \frac{1}{\hat{\lambda}} \right) < \varphi(x) - \hat{\mu}_i, \hat{q}_{ij} >_{L_2}^2 + \frac{1}{\hat{\lambda}} ||\varphi(x) - \hat{\mu}_i||_{L_2}^2 \\ &+ \sum_{j=1}^{d_i} \log(\hat{\lambda}_{ij}) + (d_{\max} - d_i) \log(\hat{\lambda}) - 2\log(\hat{\pi}_i). \end{aligned} \tag{4}$$

However, as we can see, the estimated classification function $\hat{D}_i$ still depends on the function φ and therefore requires computations in the feature space. However, since all these computations involve dot products, it will be shown in Proposition 2 that the estimated classification function can be computed without explicit knowledge of φ through a kernel function.

4.2 Estimation of the classification function through a kernel

Kernel methods are all based on the so-called "kernel trick" which allows the computation of the classifier in the observation space through a kernel K. Let us therefore introduce the kernel $K : E \times E \to \mathbb{R}$ defined as $K(x,y) =< \varphi(x), \varphi(y) >_{L_2}$ and $\rho_i : E \times E \to \mathbb{R}$ defined as $\rho_i(x,y) =< \varphi(x) - \mu_i, \varphi(y) - \mu_i >_{L_2}$. In the following, it is shown that the classification function D_i only involves ρ_i which can be computed using K:

$$\rho_i(x,y) = \frac{1}{n_i^2} \sum_{x_\ell, x_{\ell'} \in C_i} < \varphi(x) - \varphi(x_\ell), \varphi(y) - \varphi(x_{\ell'}) >_{L_2} \tag{5}$$

$$= K(x,y) - \frac{1}{n_i} \sum_{x_\ell \in C_i} (K(x_\ell, y) + K(x, x_\ell)) + \frac{1}{n_i^2} \sum_{x_\ell, x_{\ell'} \in C_i} K(x_\ell, x_{\ell'}). \tag{6}$$

Kernels	$K(x,y)$	r_i
Linear	$< x, y >_{L_2}$	$\min(n_i, p)$
RBF	$\exp\left(-\frac{\|x-y\|^2_{L_2}}{2\sigma^2}\right)$	n_i
Polynomial	$(< x, y >_{L_2} +1)^q$	$\min\left(n_i, \binom{p+q}{p}\right)$

Table 2: Dimension r_i for several classical kernels.

For each class C_i, let us introduce the $n_i \times n_i$ symmetric matrix M_i defined by:

$$(M_i)_{\ell,\ell'} = \frac{\rho_i(x_\ell, x_{\ell'})}{n_i}.$$

With these notations, we have the following result *(proof is postponed to the appendix)*.

Proposition 2. *For $i = 1, \ldots, k$, the estimated classification function can be computed, in the case of the model $\mathcal{M}_0$, as follows:*

$$\hat{D}_i(\varphi(x)) = \frac{1}{n_i}\sum_{j=1}^{d_i}\frac{1}{\hat{\lambda}_{ij}}\left(\frac{1}{\hat{\lambda}_{ij}} - \frac{1}{\hat{\lambda}}\right)\left(\sum_{x_\ell \in C_i}\beta_{ij\ell}\rho_i(x, x_\ell)\right)^2 + \frac{1}{\hat{\lambda}}\rho_i(x,x)$$
$$+ \sum_{j=1}^{d_i}\log(\hat{\lambda}_{ij}) + (d_{\max} - d_i)\log(\hat{\lambda}) - 2\log(\hat{\pi}_i),$$

where, for $j = 1, \ldots, d_i$, β_{ij} is the normed eigenvector associated to the jth largest eigenvalue $\hat{\lambda}_{ij}$ of M_i and $\hat{\lambda} = 1/\sum_{i=1}^k \hat{\pi}_i(r_i - d_i) \times \sum_{i=1}^k \hat{\pi}_i \left(\mathrm{trace}(M_i) - \sum_{j=1}^{d_i}\hat{\lambda}_{ij}\right)$.

It thus appears that each new sample point x can be assigned to the class C_i with the smallest value of the classification function without knowledge of φ. Moreover, this operation only requires the estimation of the d_i largest eigenvalues and eigenvectors of M_i. In practice, this computation is achieved thanks to the implicitly restarted Arnoldi method (IRAM) (Lehoucq and Sorensen, 1996) which is proved to be a stable procedure. The methodology based on Proposition 2 is referred to as pgpDA in the sequel. In practice, the value of r_i depends on the chosen kernel (see Table 2 for examples).

4.3 Intrinsic dimension estimation

The estimation of the intrinsic dimension of a dataset is a difficult problem which occurs frequently in data analysis, such as in principal component analysis. A classical solution in PCA is to look for a break in the eigenvalue scree of the covariance matrix. This strategy relies on the fact that the jth eigenvalue of the covariance matrix corresponds to the fraction of the full variance carried by the jth eigenvector of this matrix. Since, in our case, the class conditional matrix M_i shares with the empirical covariance operator of the associated class its largest eigenvalues, we propose to use a similar strategy based on the eigenvalue scree of the matrices M_i to estimate d_i, $i = 1, ..., k$. More precisely, we propose to make use of the scree-test of Cattell (Cattell, 1966) for estimating the class specific dimension d_i, $i = 1, ..., k$. For each class, the selected dimension is the one for which the subsequent eigenvalues differences are smaller than a threshold which can be tuned by cross-validation for instance.

4.4 Visualization in the feature subspaces

An interesting advantage of the approach is to allow the visualization of the data in subspaces of the feature space. Indeed, even though the chosen mapping function is associated with a space of very high or infinite dimension, the proposed methodology models and classifies the data in low-dimensional subspaces of the feature space. It is therefore possible to visualize the projection of the mapped data on the feature subspaces of each class using Equation (10) of the appendix. The projection of $\varphi(x)$ on the jth axis of the class C_i is therefore given by:

$$P_{ij}(\varphi(x)) := < \varphi(x) - \hat{\mu}_i, \hat{q}_{ij} >= \frac{1}{\sqrt{n_i \hat{\lambda}_{ij}}} \sum_{x_\ell \in C_i} \beta_{ij\ell} \rho_i(x, x_\ell).$$

Thus, even if the observations are non quantitative, it is possible to visualize their projections in the feature subspaces of the classes which are quantitative spaces. *Notice that the obtained visualizations of the data are of course dependent on the chosen kernel, such as in kernel PCA for instance.*

5 Links with existing works and extension to clustering

The methodology proposed in Section 4 is made very general by the large choice for the mapping function $\varphi(x)$. We focus in this section on two specific choices for $\varphi(x)$ for which the direct calculation of the classification rule is possible. We also exhibit the links with related methodologies. Finally, an extension to unsupervised classification is considered through the use of an EM algorithm.

5.1 The case of the linear kernel for quantitative data

In the case of quantitative data, $E = \mathbb{R}^p$ and one can choose $\varphi(x) = x$ associated to the standard scalar product which gives rise to the linear kernel $K(x, y) = x^t y$. In such a framework, the estimated classification function can be simplified as follows *(proof is provided in the appendix)*:

Proposition 3. *If $E = \mathbb{R}^p$ and $K(x, y) = x^t y$ then, for $i = 1, \ldots, k$, the estimated classification function reduces to*

$$\begin{aligned}\hat{D}_i(x) = &\sum_{j=1}^{d_i} \left(\frac{1}{\hat{\lambda}_{ij}} - \frac{1}{\hat{\lambda}} \right) \left(\hat{q}_{ij}^t (x - \hat{\mu}_i) \right)^2 + \frac{1}{\hat{\lambda}} ||x - \hat{\mu}_i||^2_{\mathbb{R}^p} \\ &+ \sum_{j=1}^{d_i} \log(\hat{\lambda}_{ij}) + (d_{\max} - d_i) \log(\hat{\lambda}) - 2 \log(\hat{\pi}_i).\end{aligned}$$

where $\hat{\mu}_i$ is the empirical mean of the class C_i, $\hat{q}_{ij}$ is the eigenvector of the empirical covariance matrix $\hat{\Sigma}_i$ associated to the jth largest eigenvalue $\hat{\lambda}_{ij}$ and $\hat{\lambda}$ is given by (3).

It appears that the estimated classification function reduces to the one of the high-dimensional discriminant analysis (HDDA) method (Bouveyron et al., 2007a) with the model $[a_{ij} b Q_i d]$ which has constraints similar to $\mathcal{M}_0$. Therefore, the methodology proposed in this work partially encompasses the method HDDA. Let us however highlight that the proposed methodology is not just a kernelized version of HDDA. One can remark that it is clearly not obvious to obtain Proposition 2 from Proposition 3. Indeed, substituting dot products by kernels in the classification function of the latter proposition does not yield the classification function of Proposition 2.

5.2 The case of functional data

Let us consider now functional data observed in $E = L^2([0,1])$. Let $(b_j)_{j\geq 1}$ be a basis of $L^2([0,1])$ and $F = \mathbb{R}^L$ where L is a given integer. For all $\ell = 1, \ldots, L$, the projection of a function x on the jth basis function is computed as

$$\gamma_j(x) = \int_0^1 x(t) b_j(t) dt$$

and $\gamma(x) := (\gamma_j(x))_{j=1,\ldots,L}$. Let B the $L \times L$ Gram matrix associated to the basis:

$$B_{j\ell} = \int_0^1 b_j(t) b_\ell(t) dt,$$

and consider the associated scalar product defined by $< u, v >= u^t B v$ for all $u, v \in \mathbb{R}^L$. One can then choose $\varphi(x) = B^{-1}\gamma(x)$ and $K(x,y) = \gamma(x)^t B^{-1} \gamma(y)$ leading to a simple estimated classification function.

Proposition 4. *Let $E = L^2([0,1])$ and $K(x,y) = \gamma(x)^t B^{-1}\gamma(y)$. Introduce, for $i = 1, \ldots, k$, the $L \times L$ covariance matrix of the $\gamma(x_j)$ when $x_j \in C_i$:*

$$\hat{\Sigma}_i = \frac{1}{n_i} \sum_{x_\ell \in C_i} (\gamma(x_\ell) - \bar{\gamma}_i)(\gamma(x_\ell) - \bar{\gamma}_i)^t \text{ where } \bar{\gamma}_i = \frac{1}{n_i} \sum_{x_j \in C_i} \gamma(x_j)$$

Then, for $i = 1, \ldots, k$, the estimated classification function reduces to

$$\begin{aligned}\hat{D}_i(\varphi(x)) = &\sum_{j=1}^{d_i} \left(\frac{1}{\hat{\lambda}_{ij}} - \frac{1}{\hat{\lambda}} \right) \left(\hat{q}_{ij}^t (\gamma(x) - \bar{\gamma}_i) \right)^2 + \frac{1}{\hat{\lambda}} (\gamma(x) - \bar{\gamma}_i)^t B^{-1} (\gamma(x) - \bar{\gamma}_i) \\ &+ \sum_{j=1}^{d_i} \log(\hat{\lambda}_{ij}) + (d_{\max} - d_i) \log(\hat{\lambda}) - 2\log(\hat{\pi}_i),\end{aligned}$$

where $\hat{q}_{ij}$ and $\hat{\lambda}_{ij}$ are respectively the jth normed eigenvector and eigenvalue of the matrix $B^{-1}\hat{\Sigma}_i$ and $\hat{\lambda}$ is given by (3).

Proof of the above result is provided in the appendix. Remark that $B^{-1}\hat{\Sigma}_i$ coincides with the matrix of interest in functional PCA (Ramsay and Silverman, 2005, Chap. 8.4) and that, if the basis is orthogonal, then B is the identity matrix. Notice that the proposed method therefore encompasses as well the model proposed in (Bouveyron and Jacques, 2011) for the clustering of functional data.

5.3 Links with existing works

In addition to the links with existing methods exhibited above, it is also possible to establish some links with other generative techniques proposed in the literature. Kernelized versions of Gaussian mixture models were introduced in the unsupervised (Wang et al., 2003) and supervised (Xu et al., 2009) classification contexts. These methods do not estimate the smallest eigenvalues of the kernel matrix from the data but use instead a constant threshold. Furthermore, they numerically regularize the estimated covariance matrices before to invert them. (Mika et al., 1999) have also proposed kernel Fisher discriminant (KFDA) as a non-linear version of FDA which only relies on kernel evaluations. KFDA shares as well the objectives of pgpDA but assumes a common covariance matrix for all classes and uses a ridge regularization for the inversion of this covariance matrix. Recently, pseudo-inverse and

ridge regularization have been proposed to define a kernel quadratic classifier where classes have their own covariance matrices (Pekalska and Haasdonk, 2009). Similarly, a common covariance assumption has been proposed in (Dundar and Landgrebe, 2004) for the construction of a LDA classifier in the kernel feature space. Yet, small eigenvalues are thresholded in the computations. Finally, in the unsupervised context, kernel k-means (Shawe-Taylor and Cristianini, 2004) can be also viewed as using a constraint version of our pgp models. Indeed, it uses model $\mathcal{M}_8$ with $d_i = d = r$ and $\pi_i = \pi = 1/K$ (all mixture parameters are common). To summarize, all kernel methods have to face the inversion of a kernel matrix (denoted by M_i in our framework) which is ill-conditioned or singular. The above mentioned techniques to overcome this problem include eigenvalues thresholding, shape constraints on the matrices, shrinkage methods (such as ridge regularization) and arbitrary combinations of these three methods. Here, our approach relies on the two assumptions (A1) and (A2) which benefit from clear interpretations, see Section 3.2. Moreover, no additional regularization has to be made, since the model parameters can be estimated in a stable way, see Section 4.1. Finally, the possible visualization of the data in subspaces of the feature space (Section 4.4) and the ability to deal with both supervised and non-supervised frameworks are additional arguments in favor of our approach.

5.4 Extension to unsupervised classification

In model-based classification, the unsupervised and supervised cases mainly differ in the manner to estimate the parameters of the model. The clustering task aims to form k homogeneous groups from a set of n observations $\{x_1, ..., x_n\}$ without any prior information about their group memberships. Since the labels are not available, it is not possible in this case to directly estimate the model parameters. In such a context, the expectation-maximization (EM) algorithm (Dempster et al., 1977) allows to both estimate the model parameters and predict the class memberships of the observations at hand. In the case of the parsimonious model $\mathcal{M}_0$ introduced above, the EM algorithm takes the following form:

The E step This first step reduces, at iteration s, to the computation of $t_{ij}^{(s)} = \mathbb{E}(Z_j = i|x_j, \theta^{(s-1)})$, for $j = 1, \ldots, n$ and $i = 1, \ldots, k$, conditionally on the current value of the model parameter $\theta^{(s-1)}$. *We have the following proposition (proof is postponed to the appendix):*

Proposition 5. *The conditional expectations* $t_{ij}^{(s)} = \mathbb{E}(Z_j = i|x_j, \theta^{(s-1)})$ *are given by:*

$$t_{ij}^{(s)} = 1/\sum_{\ell=1}^{k} \exp\left(D_i^{(s-1)}(\varphi(x_j)) - D_\ell^{(s-1)}(\varphi(x_j))\right), \tag{7}$$

where

$$D_i^{(s-1)}(\varphi(x)) = \frac{1}{n_i^{(s-1)}} \sum_{j=1}^{d_i} \frac{1}{\hat{\lambda}_{ij}^{(s-1)}} \left(\frac{1}{\hat{\lambda}_{ij}^{(s-1)}} - \frac{1}{\hat{\lambda}^{(s-1)}}\right) \left(\sum_{\ell=1}^{n} \beta_{ij\ell}\sqrt{t_{i\ell}}\rho_i^{(s-1)}(x, x_\ell)\right)^2$$
$$+ \frac{1}{\hat{\lambda}^{(q-1)}}\rho_i^{(s-1)}(x, x) + \sum_{j=1}^{d_i} \log(\hat{\lambda}_{ij}^{(s-1)}) + (d_{\max} - d_i)\log(\hat{\lambda}^{(s-1)}) - 2\log(\hat{\pi}_i^{(s-1)}).$$

is the Gaussian classification function associated with the model parameters estimated in the M step at iteration $s-1$ *and* $n_i^{(s-1)} = \sum_{j=1}^{n} t_{ij}^{(s-1)}$.

The M step This second step estimates the model parameters conditionally on the posterior probabilities $t_{ij}^{(q)}$ computed in the previous step. In practice, this step reduces to update the estimate of

model parameters according to the following formula:

- mixture proportions are estimated by $\hat{\pi}_i^{(q)} = n_i^{(q)}/n$ where $n_i^{(q)} = \sum_{j=1}^n t_{ij}^{(q)}$,
- parameters λ_{ij}, λ, β_{ij} and d_i are estimated at iteration q using the formula given in Proposition 2 but where the matrix M_i is now a $n \times n$ matrix, recomputed at each iteration q, and such that, for $i = 1, ..., k$ and $\ell, \ell' = 1, ..., n$:

$$\left(M_i^{(q)}\right)_{\ell,\ell'} = \frac{\sqrt{t_{i\ell}^{(q)} t_{i\ell'}^{(q)}}}{n_i^{(q)}} \rho_i^{(q)}(x_\ell, x_{\ell'})$$

  where $\rho_i^{(q)}(x_\ell, x_{\ell'})$ can be computed through the kernel K as follows:

$$\begin{aligned}\rho_i^{(q)}(x_\ell, x_{\ell'}) =& K(x_\ell, x_{\ell'}) - \frac{1}{n_i^{(q)}} \sum_{j=1}^n t_{ji}^{(q)} \left(K(x_j, x_\ell) + K(x_{\ell'}, x_j)\right) \\ &+ \frac{1}{(n_i^{(q)})^2} \sum_{j,j'=1}^n t_{ji}^{(q)} t_{j'i}^{(q)} K(x_j, x_{j'}).\end{aligned}$$

The associated clustering algorithm, denoted by pgpEM in the following, provides at convergence an estimation of the set of parameters maximizing the pseudo-likelihood associated with the parsimonious Gaussian model used. Model selection can be done using likelihood penalized criteria which add to the log-likelihood a penalty depending on the complexity of the model. Classical criteria for model selection include the AIC (Akaike, 1974), BIC (Schwarz, 1978) and ICL (Biernacki et al., 2001) criteria. The Bayesian Information Criterion (BIC) is certainly the most popular and consists in selecting the model which penalizes the likelihood by $\frac{\gamma(\mathcal{M})}{2}\log(n)$ where $\gamma(\mathcal{M})$ is the number of parameters in model $\mathcal{M}$ and n is the number of observations. On the other hand, the AIC criterion penalizes the log-likelihood by $\gamma(\mathcal{M})$ whereas the ICL criterion add the penalty $\sum_{i=1}^n \sum_{k=1}^K t_{ik}\log(t_{ik})$ to the one of the BIC criterion in order to favor well separated models. The value of $\gamma(\mathcal{M})$ is of course specific to the model selected by the practitioner.

6 Numerical experiments

In this section, numerical experiments and comparisons are conducted on simulated and real-world data sets to highlight the main features of pgpDA and pgpEM.

6.1 Simulation study: influence of d and n

A two-class non linear classification problem is first considered, see Figure 1, in order to study the influence of both the intrinsic dimension d and the size n of the data on pgpDA. The data have been simulated in $\mathbb{R}^2$ according to:

$$\begin{aligned}X_{|Z=1} &= \Big(-1 + \tau + \eta,\ 2 - \frac{\tau^2}{2} + \eta\Big), \\ X_{|Z=2} &= \Big(1 + \tau + \eta,\ -2 + \frac{\tau^2}{2} + \eta\Big),\end{aligned}$$

where $\tau \sim U[-4, 4]$ and $\eta \sim \mathcal{N}(0, 0.25)$. The first class is depicted by circles on Figure 1 whereas crosses correspond to observations of the second class. For all the experiments, *the radial basis function*

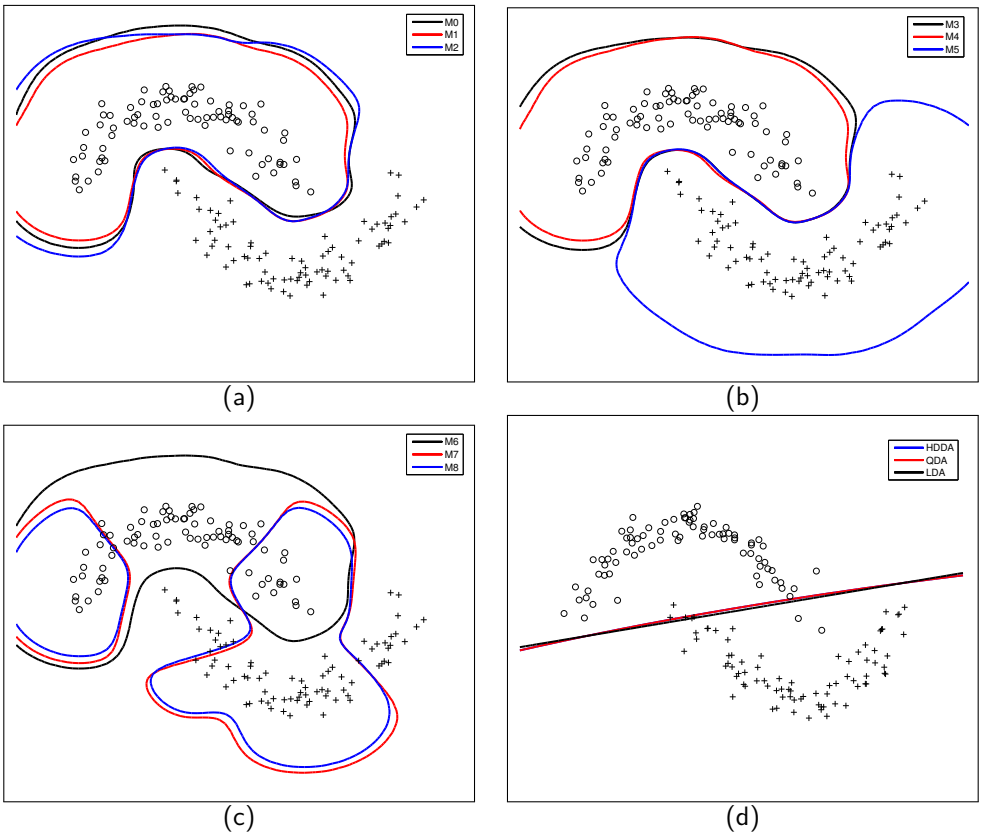


Figure 1: Decision boundaries for the simulated classification problem are depicted in color: (a) models $\mathcal{M}_0$, $\mathcal{M}_1$ and $\mathcal{M}_2$, (b) models $\mathcal{M}_3$, $\mathcal{M}_4$ and $\mathcal{M}_5$, (c) models $\mathcal{M}_6$, $\mathcal{M}_7$ and $\mathcal{M}_8$, (d) QDA, LDA and HDDA.

(RBF) kernel was used and the kernel hyper-parameter was set to $\sigma = 0.5$. For models $\mathcal{M}_0, \mathcal{M}_2$ and $\mathcal{M}_5$, the scree test threshold was fixed to 0.05. *We remind that the Cattell scree test selects, for each class, the dimension for which the subsequent eigenvalues differences are smaller than a threshold.* The decision boundaries for each pgpDA models are depicted in Figure 1. From panels (a)-(d), it appears that all the pgpDA models perform a non linear classification of the samples. For this toy data set, except $\mathcal{M}_7$ and $\mathcal{M}_8$, all the models perform similarly and the decision boundaries are very satisfying. As anyone could expect, conventional model-based classification methods (LDA, QDA, HDDA) are not be able to separate correctly those non linear data.

Let us first focus on the influence of the intrinsic dimensions on the pgpDA classifier. For the sake of simplicity, model $\mathcal{M}_1$ is considered in this experiment since it assumes a common intrinsic dimension, let us say d, for all classes. The influence of this intrinsic dimension d on the decision boundaries for model $\mathcal{M}_1$ is illustrated in Figure 2. For this experiment, we vary the intrinsic dimension value between 1 and 80. On the one hand, the parameter d seems, at least for this toy data set, not to have a strong influence since the decision boundaries are similar whatever the value of d. This behavior can be explained by the fact that all dimensions of the feature space are taken into considerations even though only the most informative are modeled precisely. On the other hand, it can be seen that when d is set to a low value, the decision boundary is slightly smoother than when it is set to a high value. Hence, we recommend the use of low values of d in practical situations in order to prevent over-fitting.

Table 3 shows the computing time for learning the pgpDA classifier according to the size of the learning set. Computing times for kernel Fisher discriminant analysis (KFD), support vector machines

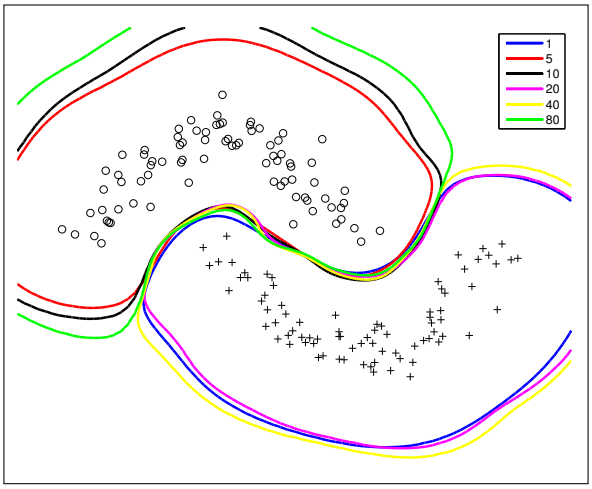


Figure 2: Simulated classification problem: Influence of the intrinsic dimension for the model $\mathcal{M}_1$. In blue $d = 1$, in red $d = 5$, in black $d = 10$, in magenta $d = 20$, in yellow $d = 40$ and in green $d = 80$.

	Learning set size				
Method	200	500	1000	1500	3000
pgpDA $\mathcal{M}_0$	0.015 ± 0.01	0.102 ± 0.01	0.473 ± 0.06	1.340 ± 0.03	12.19 ± 0.45
pgpDA $\mathcal{M}_1$	0.012 ± 0.01	0.099 ± 0.01	0.469 ± 0.06	1.336 ± 0.03	12.17 ± 0.52
pgpDA $\mathcal{M}_2$	0.013 ± 0.01	0.098 ± 0.01	0.464 ± 0.06	1.340 ± 0.03	12.31 ± 0.51
pgpDA $\mathcal{M}_3$	0.013 ± 0.01	0.098± 0.01	0.463 ± 0.06	1.332 ± 0.03	12.27 ± 0.58
pgpDA $\mathcal{M}_4$	0.012 ± 0.01	0.097 ± 0.01	0.470 ± 0.05	1.345 ± 0.05	12.15 ± 0.36
pgpDA $\mathcal{M}_5$	0.013 ± 0.01	0.098 ± 0.01	0.477 ± 0.06	1.347 ± 0.04	12.14 ± 0.33
pgpDA $\mathcal{M}_6$	0.012 ± 0.01	0.098 ± 0.01	0.479 ± 0.06	1.339 ± 0.05	12.19 ± 0.44
pgpDA $\mathcal{M}_7$	0.050 ± 0.01	0.344 ± 0.01	3.031 ± 0.13	9.453 ± 0.72	68.99 ± 2.37
pgpDA $\mathcal{M}_8$	0.045 ± 0.01	0.340 ± 0.01	3.034 ± 0.13	9.398 ± 0.39	69.58 ± 3.05
KFDA	0.005 ± 0.01	0.056 ± 0.01	0.344 ± 0.06	0.828 ± 0.02	5.26 ± 0.25
SVM	0.037 ± 0.01	0.128 ± 0.13	0.758 ± 0.09	1.904 ± 0.20	11.99 ± 1.03
GPML	0.282 ± 0.02	4.951 ± 0.65	54.38 ± 3.43	167.36 ± 13.17	1582.2 ± 128.75

Table 3: Average computing time (in seconds, ± standard deviation) over 30 replications for learning the classifiers KFDA, SVM, GPML and pgpDA according to the size of the learning set.

Dataset	n	p	n/p	k	hr
Iris	150	4	37.5	3	0.5
Glass	214	9	23.7	6	0.25
Wine	178	13	13.7	3	0.5
Ionosphere	351	34	10.3	2	0.5
Sonar	208	60	3.5	2	0.5
USPS 358	2248	256	8.8	3	0.5
Vowel	990	10	99	10	0.5
Letter	15000	16	937.5	26	0.1
Protein	*17766*	*357*	*49.8*	*3*	*0.05*
Satimage	*4435*	*36*	*123.2*	*6*	*0.25*
Vehicle	*846*	*18*	*47*	*4*	*0.5*

Table 4: Data used in the experiments. n is the number of samples, p is the number of features, k is the number of classes and hr is the hold-out ratio used in the experiments.

(SVM, with the SVM-KM toolbox, (Canu et al., 2005)) and Gaussian process classification *(GPML toolbox, (Rasmussen and Williams, 2006a))* are also provided for comparison. Notice that all codes used in this experiment are written in Matlab in order to obtain a fair comparison. *One can first observe* that pgpDA has a similar computing time as SVM. One can also remark that pgpDA has a comparable computing time as KFDA (though the latter is slightly faster), which is unsurprising since both methods are somehow related (see Section 5.3). Let us notice that models $\mathcal{M}_7$ and $\mathcal{M}_8$ of pgpDA are significantly more time consuming than the other models because their parameters (variances inside and outside the subspace, subspace orientation and intrinsic dimensions) are common between groups. As a consequence, the estimation of parameters for those models can not be done for each class independently and requires to work on a unique $n \times n$ matrix instead of several $n_i \times n_i$ matrices. Finally, the GPML implementation of Gaussian process classification turns out to be dramatically more time consuming than pgpDA, KFDA and SVM. This difference in computing times is certainly explained by the need to use the iterative expectation-propagation algorithm for inferring the GP model whereas the inference is direct for KFDA and pgpDA.

6.2 Benchmark study on quantitative data

We focus here on the comparison of pgpDA with state-of-the-art methods. To that end, two kernel generative classifiers are considered, kernel Fisher discriminant analysis (KFD) (Mika et al., 1999) and kernel LDA (KLDA) (Dundar and Landgrebe, 2004), and one kernel discriminative classifier, support vector machine (SVM) (Scholkopf and Smola, 2001). The RBF kernel is used once again in the experiments for all methods, including pgpDA. Since real-world problems are considered, all the hyper-parameters of the classifiers have been tuned using 5-fold cross-validation.

*Eleven data sets from the UCI Machine Learning Repository (*http://archive.ics.uci.edu/ml/*) have been selected: glass, ionosphere, iris, sonar, USPS, wine, vowel, letter, protein, satimage and vehicle.* We chose these data sets because they represent a wide range of situations in term of number of observations n, number of variables p and number of groups k. The USPS dataset has been modified to focus on discriminating the three most difficult classes to classify, namely the classes of the digits 3, 5 and 8. This dataset has been called USPS 358. The main feature of the data sets are described in Table 4.

Each data set was randomly split into training and testing sets in the hold-out ratio hr given in Table 4. The data were scaled between -1 and 1 on each variable. The search range for the cross-validation was

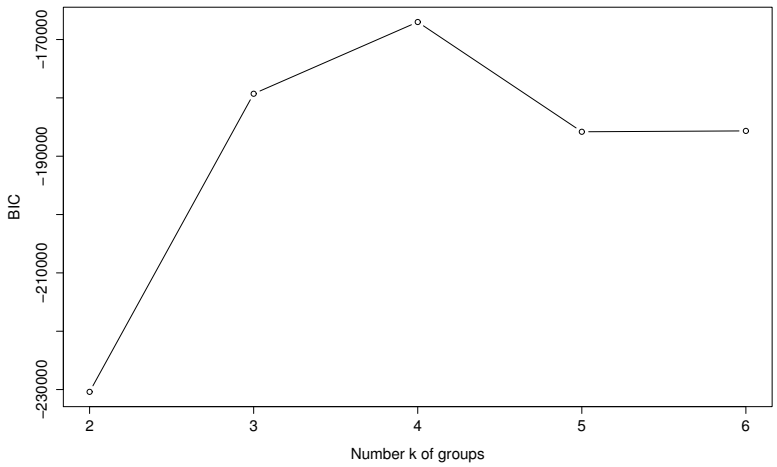


Figure 3: Choice of the number k of groups with the BIC criterion for the Canadian temperature data set.

for the kernel hyperparameter $\sigma \in [-4, 4]$, for the common intrinsic dimension $d \in [1, 20]$, for the scree test threshold $\tau \in [10^{-7}, 1]$, for the regularization parameter in KFD and KLDA $\lambda \in [10^{-13}, 10^{-6}]$ and for the penalty parameter of the SVM $\gamma \in [2^5, 2^9]$. *Notice that pgpDA has only one additional parameter to tune compared to other kernel methods: either τ for models with free dimensions or d for models with common dimensions.* The global classification accuracy was computed on the testing set and the reported results have been averaged over 50 replications of the whole process. The average classification accuracies and the standard deviations are given in Table 5.

Regarding the competitive methods, KFD and SVM provide often better results than KLDA. The model used in KLDA only fits "ionosphere", "iris" and "wine" data, for which classification accuracies are similar to or better than those obtain with KFD and SVM. For the parsimonious pgpDA models, the classification accuracies are globally good. *The results of models $\mathcal{M}_7$ and $\mathcal{M}_8$ are not presented here since they perform very poorly and we do not recommend their use. Model $\mathcal{M}_1$ provides the best result in terms of average correct classification rates. In particular, for the "USPS 358", "wine" and "satimage" data sets, it provides the best overall accuracy.* Let us remark that pgpDA performs significantly better than SVM (for the RBF kernel) on high-dimensional data (USPS 358).

In conclusion of these experiments, by relying on parsimonious models rather than regularization, pgpDA provides good classification accuracies and it is robust to the situation where few samples are available in regards to the number of variables in the original space. In practice, models $\mathcal{M}_1$ and $\mathcal{M}_4$ should be recommended: intrinsic dimension is common between the classes and the variance inside the class intrinsic subspace is either free or common. Conversely, models $\mathcal{M}_7$ and $\mathcal{M}_8$ must be avoided since they appeared to be too constrained to handle real classification situations.

6.3 Clustering of functional data: the Canadian temperatures

We now focus on illustrating the possible range of application of the proposed methodologies to different types of data. We consider here the clustering of functional data with pgpEM for which the mapping function φ is explicit (see Section 5.2). The Canadian temperature data used in this study, presented in details in (Ramsay and Silverman, 2005), consist in the daily measured temperatures at 35 Canadian

Method	Iris	Glass	Wine	Ionosphere	Sonar	USPS 358	Vowel	Letter	Protein	Satimage	Vehicle	Mean (rank)
pgpDA $\mathcal{M}_0$	**95.9± 2.1**	64.9 ± 6.3	96.8 ± 1.7	90.5 ± 2.3	77.9 ± .9	92.2 ± 1.0	**95.2 ± 1.6**	**86.5 ± 0.6**	54.8 ± 1.3	**89.4 ± 0.5**	78.3 ± 2.7	83.85 (4)
pgpDA $\mathcal{M}_1$	95.2± 2.1	62.6 ± 12.5	96.7 ± 2.3	**93.7 ± 1.6**	**81.8 ± 4.9**	**96.6 ± 0.4**	**95.2 ± 1.6**	85.4 ± 0.8	57.6 ± 0.9	**89.4 ± 0.5**	**79.9 ± 1.9**	**84.92 (1)**
pgpDA $\mathcal{M}_2$	94.4± 6.2	64.4 ± 6.7	96.8 ± 1.8	91.0 ± 2.8	71.6 ± 13.4	95.4 ± 0.8	91.2 ± 1.6	79.0 ± 0.7	54.6 ± 1.0	66.3 ± 2.5	66.8 ± 2.7	79.23 (8)
pgpDA $\mathcal{M}_3$	95.8± 2.3	64.3 ± 6.8	96.9 ± 2.0	93.2 ± 2.1	79.3 ± 4.9	96.2 ± 0.5	92.2 ± 1.5	78.7 ± 1.1	57.2 ± 1.0	77.6 ± 2.0	68.4 ± 2.3	81.80 (5)
pgpDA $\mathcal{M}_4$	94.4± 2.2	**65.3 ± 6.4**	**97.2 ± 1.8**	93.4 ± 2.0	81.6 ± 4.5	96.3 ± 0.7	95.0 ± 1.5	84.8 ± 0.7	**58.2 ± 1.3**	87.8 ± 0.9	72.8 ± 1.9	84.25 (3)
pgpDA $\mathcal{M}_5$	94.2± 7.1	59.8 ± 10.9	96.4 ± 2.0	92.0 ± 1.8	72.5 ± 12.6	96.0 ± 0.5	92.0 ± 1.5	79.0 ± 0.9	48.5 ± 1.9	63.4 ± 3.1	66.0 ± 2.9	78.16 (10)
pgpDA $\mathcal{M}_6$	94.8± 2.1	65.2 ± 5.6	**97.2 ± 1.8**	92.5 ± 2.1	79.8 ± 4.9	96.1 ± 0.5	92.4 ± 1.8	79.2 ± 1.0	56.3 ± 1.3	75.6 ± 2.3	66.9 ± 2.5	81.45 (6)
KFD	93.4± 3.7	47.3 ± 10.1	95.9 ± 2.3	**94.1 ± 1.7**	82.9 ± 3.1	**93.6 ± 0.5**	95.1 ± 1.3	83.6 ± 0.9	53.5 ± 3.8	86.1 ± 0.7	66.4 ± 12	81.08 (7)
KLDA	**96.6± 2.3**	64.5 ± 6.3	96.6 ± 1.7	88.1 ± 2.3	68.9±18.1	64.7 ± 37.5	92.4 ± 1.4	85.2 ± 0.9	41.5 ± 10.4	86.1 ± 0.7	82.1 ± 1.4	78.79 (9)
SVM	95.7± 2.0	**69.1 ± 5.5**	**96.8 ± 1.4**	92.8 ± 1.8	**84.8 ± 4.0**	77.6 ± 5.4	**96.2 ± 1.5**	**87.0 ± 0.5**	**60.7 ± 0.9**	**88.5 ± 0.5**	**82.7 ± 1.5**	**84.72 (2)**

Table 5: *Classification results on real-world datasets: reported values are average correct classification rates and standard deviation computed on validation sets.*

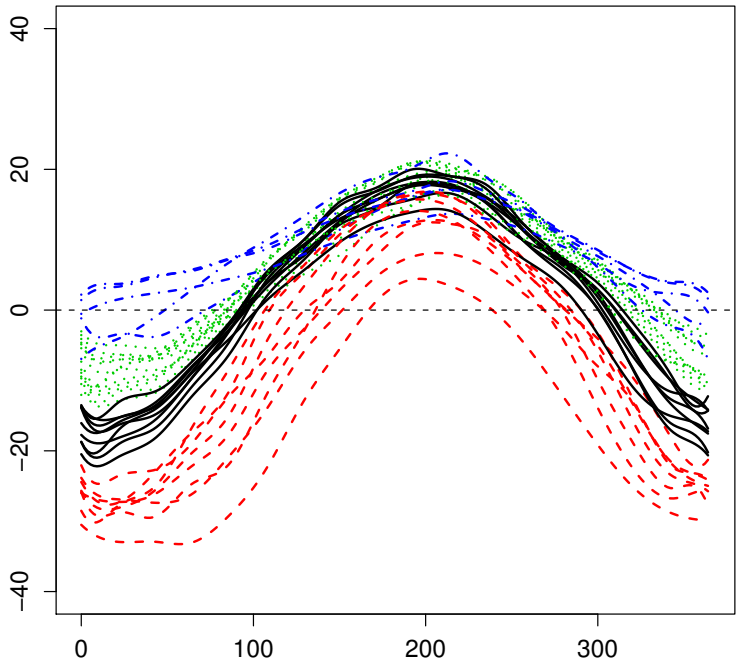


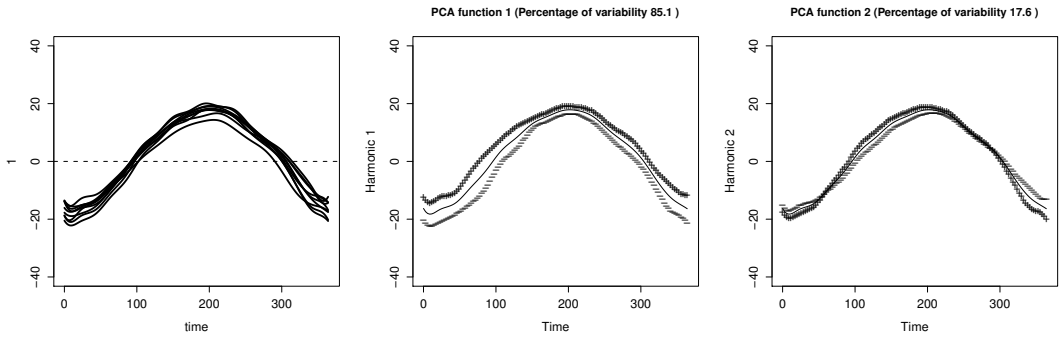
Figure 4: Clustering of the 35 times series of the Canadian temperature data set into 4 groups with pgpEM. The colors indicate the group memberships: group 1 in black, group 2 in red, group 3 in green and group 4 in blue.

weather stations across the country. The pgpEM algorithm was applied here with the model $\mathcal{M}_0$, which is the most general parsimonious Gaussian process model proposed in this work. The mapping function φ consists in the projection of the observed curves on a basis of 20 natural cubic splines. Once the pgpEM algorithm has converged, various informations are available and some of them are of particular interest. Group means, intrinsic dimensions of the group-specific subspaces and functional principal components of each group could in particular help the practitioner in understanding the clustering of the dataset at hand. *Since pgpEM is a generative clustering technique, it is possible to choose the most appropriate model for the data at hand using model selection criteria, such as BIC (Schwarz, 1978) or ICL (Biernacki et al., 2001). The number of groups has been chosen here with the BIC criterion which picks $k = 4$ (see Figure 3).* Figure 4 presents the clustering of the temperature data set into 4 groups with pgpEM.

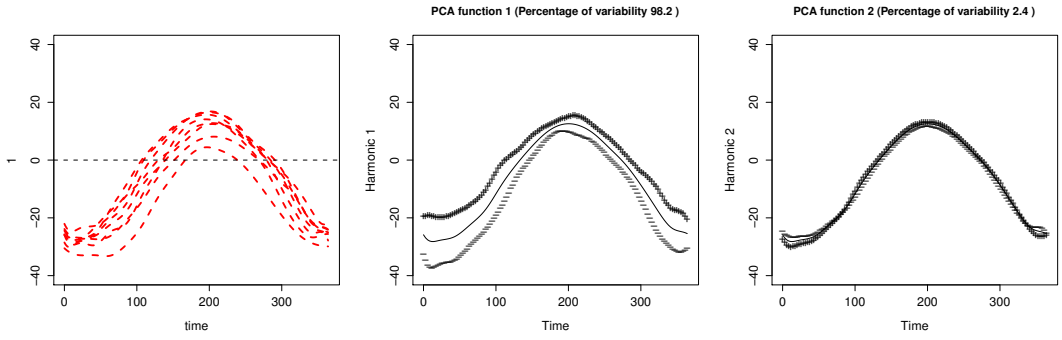
It is first interesting to have a look at the name of the weather stations gathered in the different groups formed by pgpEM. It appears that group 1 (black solid curves) is mostly made of continental stations, group 2 (red dashed curves) mostly gathers the stations of the North of Canada, group 3 (green dotted curves) mostly contains the stations of the Atlantic coast whereas the Pacific stations are mostly gathered in group 4 (blue dot-dashed curves). For instance, group 3 contains stations such as Halifax (Nova Scotia) and St Johns (Newfoundland) whereas group 4 has stations such as Vancouver and Victoria (both in British Columbia). Figure 5 provides a map of the weather stations where the colors indicate their group membership. This figure shows that the obtained clustering with pgpEM is very satisfying and rather coherent with the actual geographical positions of the stations (the clustering accuracy is 71% here compared with the geographical classification provided by (Ramsay and Silverman, 2005)). We recall that the geographical positions of the stations have not been used by pgpEM to provide the partition into 4 groups.

Figure 5: Geographical positions of the weather stations according to their group belonging. The colors indicate the group memberships: group 1 in black, group 2 in red, group 3 in green and group 4 in blue.

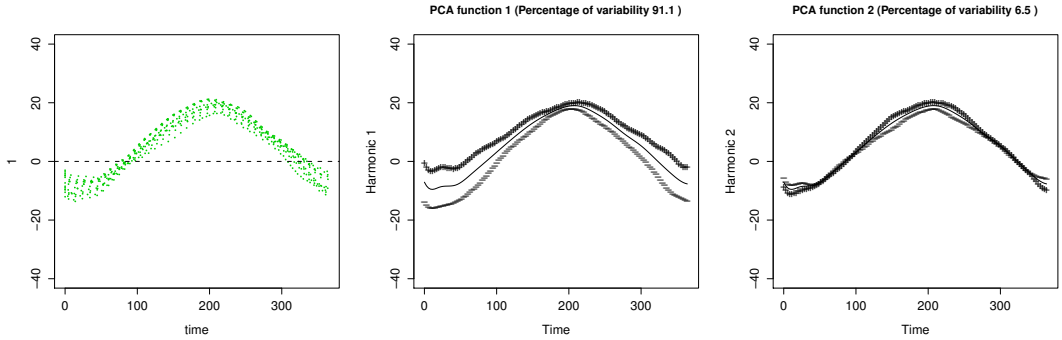
An important characteristic of the groups, but not necessarily easy to visualize, is the specific functional subspace of each group. A classical way to observe principal component functions is to plot the group mean function $\hat{\mu}_i(t)$ as well as the functions $\hat{\mu}_i(t) \pm 2\sqrt{\hat{\lambda}_{ij}}\hat{q}_{ij}(t)$ (see (Ramsay and Silverman, 2005) for more details). Figure 6 shows such a plot for the 4 groups of weather stations formed by pgpEM. *The center and right panels show the cluster mean as well as the back-projections of the perturbed mean along the two first eigenfunctions (+ and - curves). For instance, the + curve of the center panel of 1st row is the back-projection of the curve* $\hat{\mu}_1(t) + 2\sqrt{\hat{\lambda}_{11}}\hat{q}_{11}(t)$. It first appears on the first functional principal component of each group that there is more variance between the weather stations in winter than in summer. In particular, the first principal component of group 4 (blue curves, mostly Pacific stations) reveals a specific phenomenon which occurs at the beginning and the end of the winter. Indeed, we can observe a high variance in the temperatures of the Pacific coast stations at these periods of time which can be explained by the presence of mountain stations in this group. The analysis of the second principal components reveals finer phenomena. For instance, the second principal component of group 1 (black curves, mostly continental stations) shows a slight shift between the $+$ and $-$ along the year which indicates a time-shift effect. This may mean that some cities of this group have their seasons shifted, e.g. late entry and exit in the winter. Similarly, the inversion of the $+$ and $-$ on the second principal component of the Pacific and Atlantic groups (blue and green curves) suggests that, for these groups, the coldest cities in winter are also the warmest cities in summer. On the second principal component of group 2 (red curves, mostly Arctic stations), the fact that the $+$ and $-$ curves are almost superimposed shows that the North stations have very similar temperature variations (different temperature means but same amplitude) along the year.



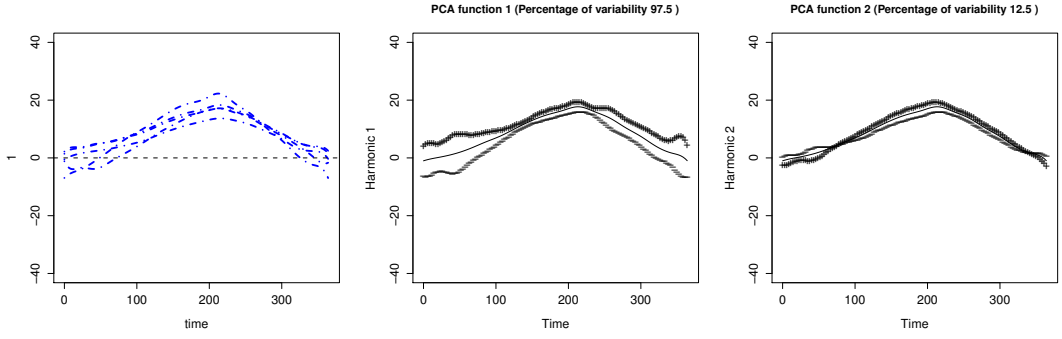
(a) Group 1 (mostly continental stations)



(b) Group 2 (mostly Arctic stations)



(c) Group 3 (mostly Atlantic stations)



(d) Group 4 (mostly Pacific stations)

Figure 6: *The group means of the Canadian temperature data obtained with pgpEM (left panel) and the effect of adding (+) or subtracting (−) twice the square root of the variance to the cluster means along the two first eigenfunctions (center and right panels, see text for details).*

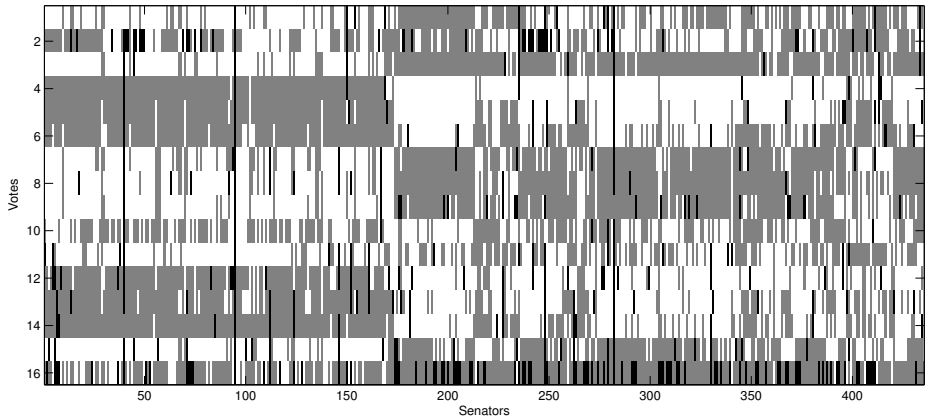


Figure 7: Votes (yea, nay or unknown) for each of the U.S. House of Representatives congressmen on 16 key votes in 1984. Yeas are in indicated in white, nays in gray and missing values in black. The first 168 congressmen are republicans whereas the 267 last ones are democrats.

6.4 Clustering of categorical data: the house-vote dataset

We focus now on categorical data which are also very frequent in scientific fields. We consider here the task of clustering (unsupervised classification) and therefore the pgpEM algorithm. To evaluate the ability of pgpEM to classify categorical data, we used the U.S. House Votes data set from the UCI repository and compare its performance with kernel k-means (Girolami, 2002; Shawe-Taylor and Cristianini, 2004). This data set is a record of the votes (yea, nay or unknown) for each of the U.S. House of Representatives congressmen on 16 key votes in 1984. These data were recorded during the third and fourth years of Ronald Reagan's Presidency. At this time, the republicans controlled the Senate, while the democrats controlled the House of Representatives. Figure 7 shows the database where yeas are in indicated in white, nays in gray and missing values in black. The first 168 congressmen are republicans whereas the 267 last ones are democrats. As we can see, the considered votes are very discriminative since republicans and democrats vote differently in almost all cases while most of the congressmen follow the majority vote in their group. We can however notice that a significant part (around 50 congressmen) of the democrats tend to vote differently from the other democrats.

To cluster this dataset, we first build a kernel from the categorical observations (16 qualitative variables with 3 possible values: yea, nay or unknown). We chose a kernel, proposed in (Couto, 2005), based on the Hamming distance which measures the minimum number of substitutions required to change one observation into another one. Naturally, pgpEM and kernel k-means worked on the same kernel to have a fair comparison and with a number of group equals to 2. The pgpEM algorithm was used with the model $\mathcal{M}_0$ and the Cattell's threshold set to 0.2 (cf. Section 4.3). Figure 8 presents the clustering result obtained with kernel k-means (left) and pgpEM (right). The clustering results are presented through a binary matrix where a black pixel indicates a common membership between two congressmen and a white pixel means different memberships for the two congressmen. The clustering accuracy between the obtained partition of the data and the democrat/republican partition was 88.97% for pgpEM and 87.59% for kernel k-means on this example. As one can observe, both algorithms globally succeed in recovering the partition of the House of Representatives. It is also interesting to notice that most of the congressmen which are not correctly classified are those who tend to vote differently from the majority vote in their group. Nevertheless, pgpEM correctly classifies 6 congressmen more than

(a) Kernel k-means (87.59% of accuracy)

(b) pgpEM (88.97% of accuracy)

Figure 8: Clustering results of the house-vote dataset obtained with kernel k-means (left) and pgpEM (right). The clustering results are presented through a binary matrix where a black pixel indicates a common membership between two congressmen. Again, the first 168 congressmen are republicans whereas the 267 last ones are democrats. On this run, pgpEM correctly classifies 6 congressmen more than kernel k-means.

kernel k-means on this run. To confirm the apparent good behavior of pgpEM, the experiment was replicated. Figure 9 presents boxplots of the clustering accuracies for kernel k-means and pgpEM over 25 replications. It turns out that pgpEM is significantly better than kernel k-means to cluster this kind of data.

6.5 Classification of mixed data: the Thyroid dataset

In this final experiment, we consider the supervised classification of mixed data which is more and more a frequent case. Indeed, it is usual to collect for the same individuals both quantitative and categorical data. For instance, in Medicine, several quantitative features can be measured for a patient (blood test results, blood pressure, morphological characteristics, ...) and these data can be completed by answers of the patient on its general health conditions (pregnancy, surgery, tabacco, ...). The Thyroid dataset considered here is from the UCI repository and contains thyroid disease records supplied by the Garavan Institute, Sydney, Australia. The dataset contains 665 records on male patients for which the answers (true of false) on 14 questions have been collected as well as 6 blood test results (quantitative measures). Among the 665 patients of the study, 61 suffer from a thyroid disease.

To make pgpDA able to deal with such data, we built a combined kernel by mixing a kernel based on the Hamming distance (Couto, 2005) for the categorical features and a RBF kernel for the quantitative data. We chose to combine both kernels simply as follows:

$$K(x_j, x_\ell) = \alpha K_1(x_j, x_\ell) + (1-\alpha) K_2(x_j, x_\ell),$$

where K_1 and K_2 are the kernels computed respectively on the categorical and quantitative features. Another solution would be to multiply both kernels. We refer to (Gönen and Alpaydin, 2011) for further details on multiple kernel learning.

The model for pgpDA was the model $\mathcal{M}_0$ with the Cattell's threshold set to 0.2 (cf. Section 4.3). *On each replication of the experiment, the data set was randomly split into a learning set of 599 observations and a test set of 66 observations.* We selected the optimal set of kernel parameters by

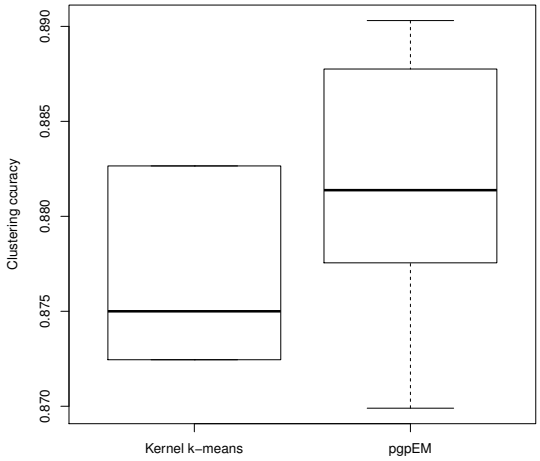


Figure 9: Boxplots of the clustering accuracies for kernel k-means (left) and pgpEM (right) over 25 replications on the house-vote dataset.

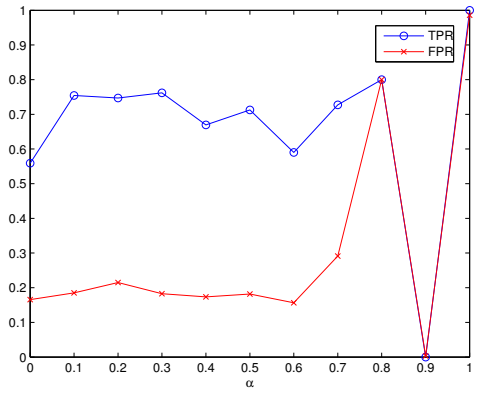
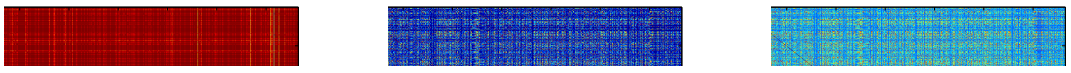


Figure 10: Choice of the mixing parameter α through cross-validation based on true positive rate (TPR) and false positive rate (FPR) curves.



Quantitative data kernel Categorical data kernel Combined kernel

Figure 11: Quantitative (left) and categorical (center) data kernels and the combined kernel (right) for the Thyroid dataset (mixed data).

Method	pgpDA on quantitative data	pgpDA on categorical data	pgpDA with the combined kernel
TP rate	74.86	96.00	75.88
FP rate	22.16	95.53	21.97

Table 6: Classification results on test sets for the Thyroid dataset (mixed data). Results are averaged on 25 replications of the experiment.

cross-validation on the learning part of the data. The average chosen parameter for the Hamming kernel was $\xi = 0.5$ ($K_1 = \exp(-D^2)/\xi^2$ where D is the Hamming distance matrix), the RBF kernel parameter was $\sigma = 15$ and the retained value for the mixing parameter α was 0.3. Figure 10 shows the choice of α and the kernels associated to the chosen values are presented in Figure 11. The rows and columns of the matrices are sorted according to the class memberships (healthy or sick) and the sick patients are the last ones. We then compared the performance of pgpDA with the combined kernel to pgpDA with, on the one hand, a simple RBF kernel built only on the quantitative variables of the dataset and, on the other hand, a Hamming kernel built only on the categorical variables. Table 6 presents both the true positive (TP) and false positive (FP) rates obtained on 25 replications of the classification experiment for pgpDA on quantitative data, on categorical data and on the mixed data. It turns out that quantitative data contains most of the important information to discriminate the patients with thyroid diseases and that categorical data, when considered alone, are not enough to build an efficient classifier. However, it appears that the use of the categorical features in combination with the quantitative data allows to slightly improve the prediction of thyroid diseases (increases the TP rate and decreases the FP rate).

7 Conclusion

This work has introduced a family of parsimonious Gaussian process models for the supervised and unsupervised classification of quantitative and non-quantitative data. The proposed parsimonious models are obtained by constraining the eigen-decomposition of the Gaussian processes modeling each class. They allow in particular to use non-linear mapping functions which project the observations into an infinite dimensional space and to build, from a finite sample, a model-based classifier in this space. It has been also demonstrated that the building of the classifier can be directly done from the observation space through a kernel, avoiding the explicit knowledge of the mapping function. It has been possible to classify data of various nature including categorical data, functional data, networks and even mixed data. The methodology is as well extended to the unsupervised classification case. Numerical experiments on benchmark data sets have shown that pgpDA performs similarly or better compared to the best kernel methods of the state of the art. The possibility to examine the model parameters and to visualize the data into the class-specific feature subspaces permits a finer interpretation of the results than with conventional discriminative kernel methods. Among the possible extensions of this work, it would be interesting to extend the methodology to the semi-supervised classification case or to the classification with uncertain labels (Bouveyron and Girard, 2009).

Acknowledgments

The authors would like to greatly thank the associate editor and the referee for their helpful remarks and comments on the manuscript.

Appendix: Proofs

Proof of Proposition 1 Recalling that $d_{\max} = \max(d_1, ..., d_k)$, the classification function can be rewritten as:

$$D_i(\varphi(x)) = \sum_{j=1}^{r} \frac{1}{\lambda_{ij}} < \varphi(x) - \mu_i, q_{ij} >^2_{L_2} + \sum_{j=1}^{d_i} \log(\lambda_{ij}) + \sum_{j=d_i+1}^{d_{\max}} \log(\lambda) - 2\log(\pi_i) + \gamma,$$

where $\gamma = (r - d_{\max})\log(\lambda)$ is a constant term which does not depend on the index i of the class. In view of the assumptions, $D_i(\varphi(x))$ can be also rewritten as:

$$\begin{aligned} D_i(\varphi(x)) &= \sum_{j=1}^{d_i} \frac{1}{\lambda_{ij}} < \varphi(x) - \mu_i, q_{ij} >^2_{L_2} + \frac{1}{\lambda} \sum_{j=d_i+1}^{r} < \varphi(x) - \mu_i, q_{ij} >^2_{L_2} \\ &+ \sum_{j=1}^{d_i} \log(\lambda_{ij}) + (d_{\max} - d_i)\log(\lambda) - 2\log(\pi_i) + \gamma. \end{aligned}$$

Introducing the norm $||.||_{L_2}$ associated with the scalar product $< ., . >_{L_2}$ and in view of Proposition 1 of (Shorack and Wellner, 1986, p. 208), we finally obtain:

$$\begin{aligned} D_i(\varphi(x)) &= \sum_{j=1}^{d_i} \left(\frac{1}{\lambda_{ij}} - \frac{1}{\lambda}\right) < \varphi(x) - \mu_i, q_{ij} >^2_{L_2} + \frac{1}{\lambda}||\varphi(x) - \mu_i||^2_{L_2} \\ &+ \sum_{j=1}^{d_i} \log(\lambda_{ij}) + (d_{\max} - d_i)\log(\lambda) - 2\log(\pi_i) + \gamma, \end{aligned}$$

which is the desired result. □

Proof of Proposition 2 The proof involves three steps.
i) Computation of the projection $< \varphi(x) - \hat{\mu}_i, \hat{q}_{ij} >_{L_2}$: Since $(\hat{\lambda}_{ij}, \hat{q}_{ij})$ is solution of the Fredholm-type equation, it follows that, for all $t \in J$,

$$\begin{aligned} \hat{\lambda}_{ij}\hat{q}_{ij}(t) &= \int_J \hat{\Sigma}_i(s,t)\hat{q}_{ij}(s)ds \\ &= \frac{1}{n_i} \sum_{x_\ell \in C_i} < \varphi(x_\ell) - \hat{\mu}_i, \hat{q}_{ij} >_{L_2} (\varphi(x_\ell)(t) - \hat{\mu}_i(t)). \end{aligned} \quad (8)$$

This implies that $\hat{q}_{ij}$ lies in the linear subspace spanned by the $(\varphi(x_\ell) - \hat{\mu}_i)$, $x_\ell \in C_i$. As a consequence, the rank of the operator $\hat{\Sigma}_i$ is finite and is at most $r_i = \min(n_i, r)$. It therefore exists $\beta_{ij\ell} \in \mathbb{R}$ such that:

$$\hat{q}_{ij} = \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}} \sum_{x_\ell \in C_i} \beta_{ij\ell}(\varphi(x_\ell) - \hat{\mu}_i) \quad (9)$$

leading to:

$$< \varphi(x) - \hat{\mu}_i, \hat{q}_{ij} >_{L_2} = \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}} \sum_{x_\ell \in C_i} \beta_{ij\ell}\rho_i(x, x_\ell), \quad (10)$$

for all $j = 1, \ldots, r_i$. The estimated classification function has therefore the following form:

$$\hat{D}_i(\varphi(x)) = \frac{1}{n_i}\sum_{j=1}^{d_i}\frac{1}{\hat{\lambda}_{ij}}\left(\frac{1}{\hat{\lambda}_{ij}}-\frac{1}{\hat{\lambda}}\right)\left(\sum_{x_\ell\in C_i}\beta_{ij\ell}\rho_i(x,x_\ell)\right)^2+\frac{1}{\hat{\lambda}}\rho_i(x,x)$$
$$+\sum_{j=1}^{d_i}\log(\hat{\lambda}_{ij})+(d_{\max}-d_i)\log(\hat{\lambda})-2\log(\hat{\pi}_i),$$

for all $i = 1, \ldots, k$.

ii) Computation of the $\beta_{ij\ell}$ and $\hat{\lambda}_{ij}$: Replacing (9) in the Fredholm-type equation (8) it follows that

$$\frac{1}{n_i}\sum_{x_\ell,x_{\ell'}\in C_i}\beta_{ij\ell'}(\varphi(x_\ell)-\hat{\mu}_i)\rho_i(x_\ell,x_{\ell'})=\hat{\lambda}_{ij}\sum_{x_{\ell'}\in C_i}\beta_{ij\ell'}(\varphi(x_{\ell'})-\hat{\mu}_i).$$

Finally, projecting this equation on $\varphi(x_m)-\hat{\mu}_i$ for $x_m \in C_i$ yields

$$\frac{1}{n_i}\sum_{x_\ell,x_{\ell'}\in C_i}\beta_{ij\ell'}\rho_i(x_\ell,x_m)\rho_i(x_\ell,x_{\ell'})=\hat{\lambda}_{ij}\sum_{x_{\ell'}\in C_i}\beta_{ij\ell'}\rho_i(x_m,x_{\ell'}).$$

Recalling that M_i is the matrix $n_i \times n_i$ defined by $(M_i)_{\ell,\ell'} = \rho_i(x_\ell, x_{\ell'})/n_i$ and introducing β_{ij} the vector of $\mathbb{R}^{n_i}$ defined by $(\beta_{ij})_\ell = \beta_{ij\ell}$, the above equation can be rewritten as $M_i^2\beta_{ij} = \hat{\lambda}_{ij}M_i\beta_{ij}$ or, after simplification $M_i\beta_{ij} = \hat{\lambda}_{ij}\beta_{ij}$. As a consequence, $\hat{\lambda}_{ij}$ is the jth largest eigenvalue of M_i and β_{ij} is the associated eigenvector for all $1 \le j \le d_i$. Let us note that the constraint $\|\hat{q}_{ij}\| = 1$ can be rewritten as $\beta_{ij}^t\beta_{ij} = 1$.

iii) Computation of $\hat{\lambda}$: Remarking that $\text{trace}(\hat{\Sigma}_i) = \text{trace}(M_i) + \sum_{j=r_i+1}^{r}\hat{\lambda}_{ij}$, it follows:

$$\hat{\lambda}=\frac{1}{\sum_{i=1}^k\hat{\pi}_i(r_i-d_i)}\sum_{i=1}^k\hat{\pi}_i\left(\text{trace}(M_i)-\sum_{j=1}^{d_i}\hat{\lambda}_{ij}\right),$$

and the proposition is proved. □

Proof of Proposition 3 It is sufficient to prove that $\hat{q}_{ij}$ and $\hat{\lambda}_{ij}$ are respectively the jth normed eigenvector and eigenvalue of $\hat{\Sigma}_i$. First,

$$\begin{aligned}
\hat{\Sigma}_i\hat{q}_{ij} &= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}\frac{1}{n_i}\sum_{x_{\ell'}\in C_i}(x_{\ell'}-\bar{\mu}_i)(x_{\ell'}-\bar{\mu}_i)^t\sum_{x_\ell\in C_i}\beta_{ij\ell}(x_\ell-\bar{\mu}_i)\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}\frac{1}{n_i}\sum_{x_{\ell'},x_\ell\in C_i}\beta_{ij\ell}(x_{\ell'}-\bar{\mu}_i)(x_{\ell'}-\bar{\mu}_i)^t(x_\ell-\bar{\mu}_i)\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}\frac{1}{n_i}\sum_{x_{\ell'},x_\ell\in C_i}\beta_{ij\ell}(x_{\ell'}-\bar{\mu}_i)\rho_i(x_\ell,x_{\ell'})\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}\sum_{x_{\ell'},x_\ell\in C_i}(M_i)_{\ell,\ell'}\beta_{ij\ell}(x_{\ell'}-\bar{\mu}_i)\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\sum_{x_{\ell'}\in C_i}(M_i\beta_{ij})_{\ell'}(x_{\ell'}-\bar{\mu}_i),
\end{aligned}$$

and remarking that β_{ij} is eigenvector of M_i, it follows:

$$\hat{\Sigma}_i \hat{q}_{ij} = \hat{\lambda}_{ij} \frac{1}{\sqrt{n_i \hat{\lambda}_{ij}}} B^{-1} \sum_{x_{\ell'} \in C_i} \beta_{ij\ell'}(x_{\ell'} - \bar{\mu}_i) = \hat{\lambda}_{ij} \hat{q}_{ij}.$$

Second, straightforward algebra shows that

$$\begin{aligned}
||\hat{q}_{ij}||^2 &= \frac{1}{n_i \hat{\lambda}_{ij}} \sum_{x_\ell \in C_i} \beta_{ij\ell}(x_\ell) - \bar{\mu}_i)^t \sum_{x_{\ell'} \in C_i} \beta_{ij\ell'}(x_{\ell'} - \bar{\mu}_i) \\
&= \frac{1}{n_i \hat{\lambda}_{ij}} \sum_{x_{\ell'}, x_\ell \in C_i} \beta_{ij\ell} \beta_{ij\ell'}(x_\ell - \bar{\mu}_i)^t (x_{\ell'} - \bar{\mu}_i) \\
&= \frac{1}{\hat{\lambda}_{ij}} \sum_{x_{\ell'}, x_\ell \in C_i} (M_i)_{\ell,\ell'} \beta_{ij\ell} \beta_{ij\ell'} \\
&= \frac{1}{\hat{\lambda}_{ij}} \hat{q}_{ij}^t M_i \hat{q}_{ij} = 1,
\end{aligned}$$

and the result is proved. □

Proof of Proposition 4 For all $\ell = 1, \ldots, L$, the ℓth coordinate of the mapping function $\varphi(x)$ is defined as the ℓth coordinate of the function x expressed in the truncated basis $\{b_1, \ldots, b_L\}$. More specifically,

$$x(t) = \sum_{\ell=1}^{L} \varphi_\ell(x) b_\ell(t),$$

for all $t \in [0, 1]$ and thus, for all $j = 1, \ldots, L$, we have

$$\gamma_j(x) = \int_0^1 x(t) b_j(t) dt = \sum_{\ell=1}^{L} \varphi_\ell(x) \int_0^1 b_j(t) b_\ell(t) dt = \sum_{\ell=1}^{L} B_{j\ell} \varphi_\ell(x).$$

As a consequence, $\varphi(x) = B^{-1}\gamma(x)$ and $K(x, y) = \gamma(x)^t B^{-1} \gamma(y)$. Introducing

$$\bar{\gamma}_i = \frac{1}{n_i} \sum_{x_j \in C_i} \gamma(x_j),$$

it follows that $\rho_i(x, y) = (\gamma(x) - \bar{\gamma}_i)^t B^{-1} (\gamma(y) - \bar{\gamma}_i)$. Let us first show that $\hat{q}_{ij}$ is eigenvector of $B^{-1}\hat{\Sigma}_i$. Recalling that

$$\hat{q}_{ij} = \frac{1}{\sqrt{n_i \hat{\lambda}_{ij}}} B^{-1} \sum_{x_\ell \in C_i} \beta_{ij\ell}(\gamma(x_\ell) - \bar{\gamma}_i),$$

we have

$$
\begin{aligned}
B^{-1}\hat{\Sigma}_i\hat{q}_{ij} &= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\frac{1}{n_i}\sum_{x_{\ell'}\in C_i}(\gamma(x_{\ell'})-\bar{\gamma}_i)(\gamma(x_{\ell'})-\bar{\gamma}_i)^tB^{-1}\sum_{x_\ell\in C_i}\beta_{ij\ell}(\gamma(x_\ell)-\bar{\gamma}_i)\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\frac{1}{n_i}\sum_{x_{\ell'},x_\ell\in C_i}\beta_{ij\ell}(\gamma(x_{\ell'})-\bar{\gamma}_i)(\gamma(x_{\ell'})-\bar{\gamma}_i)^tB^{-1}(\gamma(x_\ell)-\bar{\gamma}_i)\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\frac{1}{n_i}\sum_{x_{\ell'},x_\ell\in C_i}\beta_{ij\ell}(\gamma(x_{\ell'})-\bar{\gamma}_i)\rho_i(x_\ell,x_{\ell'})\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\sum_{x_{\ell'},x_\ell\in C_i}(M_i)_{\ell,\ell'}\beta_{ij\ell}(\gamma(x_{\ell'})-\bar{\gamma}_i)\\
&= \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\sum_{x_{\ell'}\in C_i}(M_i\beta_{ij})_{\ell'}(\gamma(x_{\ell'})-\bar{\gamma}_i).
\end{aligned}
$$

Remarking that β_{ij} is eigenvector of M_i, it follows:

$$
B^{-1}\hat{\Sigma}_i\hat{q}_{ij} = \hat{\lambda}_{ij}\frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}B^{-1}\sum_{x_{\ell'}\in C_i}\beta_{ij\ell'}(\gamma(x_{\ell'})-\bar{\gamma}_i) = \hat{\lambda}_{ij}\hat{q}_{ij}.
$$

Let us finally compute the norm of $\hat{q}_{ij}$:

$$
\begin{aligned}
||\hat{q}_{ij}||^2 &= \frac{1}{n_i\hat{\lambda}_{ij}}\sum_{x_\ell\in C_i}\beta_{ij\ell}(\gamma(x_\ell)-\bar{\gamma}_i)^tB^{-1}\sum_{x_{\ell'}\in C_i}\beta_{ij\ell'}(\gamma(x_{\ell'})-\bar{\gamma}_i)\\
&= \frac{1}{n_i\hat{\lambda}_{ij}}\sum_{x_{\ell'},x_\ell\in C_i}\beta_{ij\ell}\beta_{ij\ell'}(\gamma(x_\ell)-\bar{\gamma}_i)^tB^{-1}(\gamma(x_{\ell'})-\bar{\gamma}_i)\\
&= \frac{1}{\hat{\lambda}_{ij}}\sum_{x_{\ell'},x_\ell\in C_i}(M_i)_{\ell,\ell'}\beta_{ij\ell}\beta_{ij\ell'}\\
&= \frac{1}{\hat{\lambda}_{ij}}\hat{q}_{ij}^tM_i\hat{q}_{ij} = 1,
\end{aligned}
$$

and the result is proved. □

Proof of Proposition 5 *Let us first set the estimate of $\mu_i$at iteration s to its empirical counterpart conditionally on the current value of the model parameter $\theta^{(s-1)}$:*

$$
\hat{\mu}_i^{(s)}(t) = \frac{1}{n_i^{(s)}}\sum_{j=1}^n t_{ij}^{(s-1)}\varphi(x_j)(t),
$$

where $n_i^{(s-1)} = \sum_{j=1}^n t_{ij}^{(s-1)}$. Replacing $\hat{\mu}_i$ by $\hat{\mu}_i^{(s)}$ in the proof of Proposition 2 , we get:

$$
\hat{q}_{ij} = \frac{1}{\sqrt{n_i\hat{\lambda}_{ij}}}\sum_{x_\ell\in C_i}\beta_{ij\ell}\sqrt{t_{\ell i}}(\varphi(x_\ell)-\hat{\mu}_i). \tag{11}
$$

According to Bayes' rule and Equation (1), the computation of $t_{ij}^{(s)} = \mathbb{E}(Z_j = i|x_j, \theta^{(s-1)})$ *conditionally on the current value of the model parameter* $\theta^{(s-1)}$ *can be finally written as:*

$$t_{ij}^{(s)} = \mathbb{E}(Z_j = i|x_j, \theta^{(s-1)}) = P(Z_j = i|x_j, \theta^{(s-1)}) = \frac{\exp\left(-\frac{1}{2}D_i^{(s-1)}\left(\varphi(x_j)\right)\right)}{\sum_{\ell=1}^{k}\exp\left(-\frac{1}{2}D_\ell^{(s-1)}\left(\varphi(x_j)\right)\right)},$$

for $j = 1, \ldots, n$ *and* $i = 1, \ldots, k$*. The result follows.* □

References

Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

J.L. Andrews and P.D. McNicholas. Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Statistics and Computing*, 22(5):1021–1029, 2012.

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2001.

N. Bouguila, D. Ziou, and J. Vaillancourt. Novel mixtures based on the Dirichlet distribution: application to data and image classification. In *Machine Learning and Data Mining in Pattern Recognition*, pages 172–181. Springer, 2003.

C. Bouveyron and C. Brunet. Simultaneous model-based clustering and visualization in the Fisher discriminative subspace. *Statistics and Computing*, 22(1):301–324, 2012.

C. Bouveyron and C. Brunet-Saumard. Model-based clustering of high-dimensional data: A review. *Computational Statistics and Data Analysis*, 71:52–78, 2013.

C. Bouveyron and S. Girard. Robust supervised classification with mixture models : Learning from data with uncertain labels. *Pattern Recognition*, 42(11):2649–2658, 2009.

C. Bouveyron and J. Jacques. Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300, 2011.

C. Bouveyron, S. Girard, and C. Schmid. High-dimensional discriminant analysis. *Communication in Statistics: Theory and Methods*, 36:2607–2623, 2007a.

C. Bouveyron, S. Girard, and C. Schmid. High-dimensional data clustering. *Computational Statistics and Data Analysis*, 52:502–519, 2007b.

S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. SVM and kernel methods matlab toolbox. Perception Systemes et Information, INSA de Rouen, Rouen, France, 2005.

A. Caponnetto, C.A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 68:1615–1646, 2008.

R. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966.

G. Celeux and G. Govaert. Clustering criteria for discrete data and latent class models. *Journal of classification*, 8(2):157–176, 1991.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

J. Couto. Kernel k-means for categorical data. In *Advances in Intelligent Data Analysis VI*, volume 3646 of *Lecture Notes in Computer Science*, pages 739–739. 2005.

M. Cuturi and J.P. Vert. The context-tree kernel for strings. *Neural Networks*, 18(8):1111–1123, 2005.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

M.M. Dundar and D.A. Landgrebe. Toward an optimal supervised classifier for the analysis of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 42(1):271 – 277, 2004.

T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

R.A. Fisher. The use of multiple measurements in taxonomic problems. *The Annals of Eugenics*, 7: 179–188, 1936.

F. Forbes and D. Wraith. A new family of multivariate heavy-tailed distributions with variable marginal amounts of tailweight: application to robust clustering. *Statistics and Computing*, to appear, 2014.

B.C. Franczak, R.P. Browne, and P.D. McNicholas. Mixtures of shifted asymmetric Laplace distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1149–1157, 2014.

M. Girolami. Mercer kernel-based clustering in feature space. *Neural Networks, IEEE Transactions on*, 13(3):780–784, 2002.

M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

T. Hofmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

H. Kadri, A. Rakotomamonjy, F. Bach, and P. Preux. Multiple Operator-valued Kernel Learning. In *Neural Information Processing Systems (NIPS)*, pages 1172–1080, 2012.

M. Kuss and C. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

S. Lee and G.J. McLachlan. Finite mixtures of multivariate skew t-distributions: some recent and new results. *Statistics and Computing*, 24(2):181–202, 2013.

R. Lehoucq and D. Sorensen. Deflation techniques for an implicitly restarted arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

T.I. Lin. Robust mixture modeling using multivariate skew t distribution. *Statistics and Computing*, 20:343–356, 2010.

T.I. Lin, J.C. Lee, and W.J. Hsieh. Robust mixture modeling using the skew t distribution. *Statistics and Computing*, 17:81–92, 2007.

P. Mahé and J.P. Vert. Graph kernels based on tree patterns for molecules. *Machine learning*, 75(1): 3–35, 2009.

G. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.

G. McLachlan, D. Peel, and R. Bean. Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics and Data Analysis*, 41:379–388, 2003.

P. McNicholas and B. Murphy. Parsimonious Gaussian mixture models. *Statistics and Computing*, 18 (3):285–296, 2008.

S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K.R. Müllers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing (NIPS)*, pages 41–48, 1999.

T. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

A. Montanari and C. Viroli. Heteroscedastic factor mixture analysis. *Statistical Modeling: An International journal*, 10(4):441–460, 2010.

T.B. Murphy, N. Dean, and A.E. Raftery. Variable selection and updating in model-based discriminant analysis for high dimensional data with food authenticity applications. *The Annals of Applied Statistics*, 4(1):219–223, 2010.

A. Murua and N. Wicker. Kernel-based Mixture Models for Classification. Technical report, University of Montréal, 2014.

E. Pekalska and B. Haasdonk. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1017 –1032, june 2009.

J. O. Ramsay and B. W. Silverman. *Functional data analysis*. Springer Series in Statistics. Springer, New York, second edition, 2005.

C. Rasmussen and C. Williams. Gaussian processes for machine learning matlab toolbox. The MIT press, 2006a.

C. Rasmussen and C. Williams. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006b.

B. Scholkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

B. Schölkopf, A. Smola, and K-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

B. Schölkopf, K. Tsuda, and J.-P. Vert, editors. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

G.R. Shorack and J.A. Wellner. *Empirical Processes with Applications to Statistics*. Wiley, New York, 1986.

A. Smola and R. Kondor. Kernels and regularization on graphs. In *Proc. Conf. on Learning Theory and Kernel Machines*, pages 144–158, 2003.

J. Wang, J. Lee, and C. Zhang. Kernel trick embedded Gaussian mixture model. In *Proceedings of the 14th international conference on algorithmic learning theory*, pages 159–174, 2003.

Z. Xu, K. Huang, J. Zhu, I. King, and M.R. Lyu. A novel kernel-based maximum a posteriori classification method. *Neural Networks*, 22:977–987, 2009.