Cluster-Specific Predictions with Multi-Task Gaussian Processes

Arthur Leroy, Pierre Latouche, Benjamin Guedj, Servane Gey

Cluster-Specific Predictions with Multi-Task Gaussian Processes

Arthur Leroy

Department of Computer Science, The University of Manchester, United Kingdom
arthur.leroy.pro@gmail.com

Pierre Latouche

Université de Paris, CNRS, MAP5 UMR 8145, F-75006 Paris, France
pierre.latouche@u-paris.fr

Benjamin Guedj

Inria, France and University College London, United Kingdom
benjamin.guedj@inria.fr

Servane Gey

Université de Paris, CNRS, MAP5 UMR 8145, F-75006 Paris, France
servane.gey@u-paris.fr

July 29, 2022

A model involving Gaussian processes (GPs) is introduced to simultaneously handle multi-task learning, clustering, and prediction for multiple functional data. This procedure acts as a model-based clustering method for functional data as well as a learning step for subsequent predictions for new tasks. The model is instantiated as a mixture of multi-task GPs with common mean processes. A variational EM algorithm is derived for dealing with the optimisation of the hyper-parameters along with the hyper-posteriors' estimation of latent

variables and processes. We establish explicit formulas for integrating the mean processes and the latent clustering variables within a predictive distribution, accounting for uncertainty on both aspects. This distribution is defined as a mixture of cluster-specific GP predictions, which enhances the performances when dealing with group-structured data. The model handles irregular grids of observations and offers different hypotheses on the covariance structure for sharing additional information across tasks. The performances on both clustering and prediction tasks are assessed through various simulated scenarios and real data sets. The overall algorithm, called MagmaClust, is publicly available as an R package.

Keywords. Gaussian processes mixture, curve clustering, multi-task learning, variational EM, cluster-specific predictions

1 Introduction

The classic context of regression aims at inferring the underlying mapping function associating input to output data. In a probabilistic framework, some strategies assume that this function is drawn from a prior Gaussian process (GP). According to **?**, a Gaussian process can be defined as a collection of random variables (indexed over a continuum), any finite number of which having a joint Gaussian distribution. From this definition, we may enforce some properties for the target function solely by characterising the mean and covariance parameters of the process. Since GPs are an example of kernel methods, a broad range of assumptions can be expressed through the definition of the covariance function. We refer to **?** for a comprehensive review. Despite undeniable advantages, natural implementations for GPs scale cubically with the number of data points, which constitutes a major drawback in many applications. Thereby, the early literature focused on deriving tractable approximations to mitigate this problem (**????**). Subsequent reviews (**??**) have also provided standardised formulations and comparisons on this topic. Besides, several approximations have been proposed (**?**) and implemented (**??**) for tackling the issue of non-Gaussian errors and adapting GPs to a broad variety of likelihoods. Since a GP corresponds to a probability distribution over a functional space, alternate approaches for modelling functional data (**?**) should also be mentioned, in particular for our clustering purpose.

Unsupervised learning of functional data, also called *curve clustering*, focuses on the definition of sub-groups of curves, making sense according to an appropriate measure of similarity. When dealing with functional data, the concept of basis functions expansion is of paramount importance for defining smooth and manageable representations of the data. Such a representation allows the adaptation of multivariate methods such as k-means, in combination with B-splines bases for instance (**?**), to handle curve clustering problems. Different bases can be used, such as Fourier or wavelets (**?**), according to the context and the nature of the signal. Besides, model-based clustering methods aims at defining probabilistic techniques for this task, and many approaches (**??**) have been proposed in this sense for the past decade. In particular, the algorithms *funHDDC* (**?**) and *funFEM* (**?**) establish a mixture model where representative coefficients of the curves are supposed to come from cluster-specific Gaussian distributions. Furthermore, the authors in **?** introduced an extension to the case of multivariate functional data. A comprehensive review (**?**) has been proposed to discuss and compare the major approaches of this active research

area. We can also mention recent works leveraging generalised Bayesian predictors and PAC-Bayes for learning and clustering streams of data (**??**), which later inspired a work on clustering (**?**). In line with the previous methods that simultaneously analyse multiple curves, we also introduce a framework that takes advantage of similarities between resembling data.

The *multi-task* paradigm consists in using data from several *tasks* (also called *batches* or *individuals*) to improve the learning or predictive capacities compared to an isolated model. It was introduced by **?** and then adapted in many fields of machine learning. An initial GP adaptation (**?**) came as a hierarchical Bayesian model using an expectation-maximisation (EM) algorithm for learning, and a similar approach can be found in **?**. Another hierarchical formulation using a GP to model the mean parameter of another GP was later proposed in **?**. Such modelling assumptions resemble those of the present paper, although the strategies used for learning and prediction largely differ. Meanwhile, **?** offered an extensive study of the relationships between the linear model and GPs to develop a multi-task formulation. More recently, the expression *multi-task GP* has been coined by **?** for referring to a covariance structure involving inputs and tasks in two separate matrices. Some further developments on this approach were proposed (**???**) and we can also mention the work of **?** on Bayesian hyper-parameter optimisation in such models. Generally, as presented in the review **?** which favours the term *multi-output GP*, all these frameworks can be expressed as specific cases of the *linear model of coregionalisation* (LMC) introduced by **?** in the field of geostatistics. Finally, let us emphasise the algorithm Magma from **?** that recently proposed a different multi-task paradigm for GPs, by transferring information through a latent mean process rather than the covariance structure, an intuition that partially appeared before in **?**. This approach offers enhanced performances in forecasting while scaling linearly in the number of tasks, which is noticeably lower than the previous multi-output methods which generally bear a cubic complexity. However, the assumption of a unique mean process might happen to be too restrictive and could benefit from a more general formulation. For instance, **?** proposed an idea close to our following model by introducing a curve clustering component to a hybrid splines-GPs multi-task framework, although their approach does not fully account for uncertainty and cannot handle irregular measurements. Moreover, no implementation has been released for their algorithm, and by deriving a unified multi-task GP framework that is more general, we aim at offering to practitioners a powerful tool for tackling the simultaneous clustering and prediction of multiple functional data.

Our contributions. The present paper contributes a significant extension of Magma (**?**), by introducing a clustering component into the procedure. To this end, **(i)** we introduce a more general model involving multiple mean GPs, each one being associated with a particular cluster. These processes represent the prior mean trend, possibly different from one cluster to another, that is associated with an individual covariance structure for each functional data. Moreover, we propose 4 different modelling hypotheses regarding the kernels' hyper-parameters of the GPs. **(ii)** We derive a variational expectation-maximisation (VEM) algorithm called MagmaClust (available as an R package on the CRAN and at https://github.com/ArthurLeroy/MagmaClustR) for estimating the hyper-parameters along with the hyper-posterior distributions of the mean processes and latent clustering variables. A variational BIC criterion is proposed to estimate the number of clusters. **(iii)** We enrich this learning step with an additional EM algorithm and analytical formulas to

determine both clusters probabilities and predictive distributions for any new, partially observed, individual. The final multi-task prediction can be expressed in terms of cluster-specific distributions or as an overall GPs mixture. The algorithmic complexity of learning and prediction steps are discussed as well. **(iv)** We illustrate the advantages of our approach on synthetic and three real-life data sets. The results exhibit that MagmaClust outperforms state-of-the-art alternatives on both curve clustering and prediction tasks, in particular for group-structured data sets.

Outline of the paper. We introduce the multi-task Gaussian processes mixture model in **??**, along with notation. **??** is devoted to the inference procedure, with a Variational Expectation-Maximisation (VEM) algorithm to estimate hyper-parameters and approximation of hyper-posterior distributions along with mixture proportions. We leverage this strategy in **??** and derive both a mixture and cluster-specific GP prediction formulas, for which we provide an analysis along with computational costs in **??**. The performances of our algorithm for clustering and prediction purposes are illustrated in **??** with a series of experiments on both synthetic and real-life data sets along with comparisons to competing state-of-the-art algorithms. We close with a summary of our work in **??**. All proofs to original results are deferred to **??**.

2 Modelling

2.1 Motivation

Before diving into modelling considerations, let us provide a motivational example used throughout the article to illustrate the kind of problems we expect to tackle. Assume that we have observed results of swimming competitions for thousands of athletes from 10 to 20 years old. In order to improve talent detection, one might be interested in using those data to forecast future performances for new young swimmers (e.g observed only between 10 and 14 years old). Such a data set is composed of thousands of irregular age-performance time series, where each swimmer would have specific number and locations of their data points, as illustrated in **??**. These examples come from a real data set, thoroughly described in **??**, which was originally the applicative motivation for developing methods described in the present paper. The following multi-task GPs framework is tailored to simultaneously allocate individuals into clusters while learning model parameters, and then provide probabilistic predictions for future performances of any young swimmer, by sharing information between resembling individuals through common mean trends.

2.2 Notation

To remain consistent with this illustrative example, we refer throughout to the input variables as *timestamps* and use the term *individual* as a synonym of batch or task. However, although the temporal formulation helps intuition, the present framework still applies to the wide range of inputs one can usually think of in GP applications. As we suppose the data set to be composed of point-wise observations from multiple functions, the set of all indices is denoted by $\mathcal{I} \subset \mathbb{N}$, which in particular contains $\{1, \dots, M\}$, the indices of the observed individuals (i.e. the training set). Since the input values are defined over a continuum, let us name $\mathcal{T}$ this input space (we can assume $\mathcal{T} \subset \mathbb{R}$ here for simplicity). Moreover, since the following model is defined for clustering purposes, the set of indices $\mathcal{K} = \{1, \dots, K\}$ refers to the K different groups of individuals. For the sake of concision, the notation is shortened as follows: for any object x, $\{x_i\}_i = \{x_1, \dots, x_M\}$ and

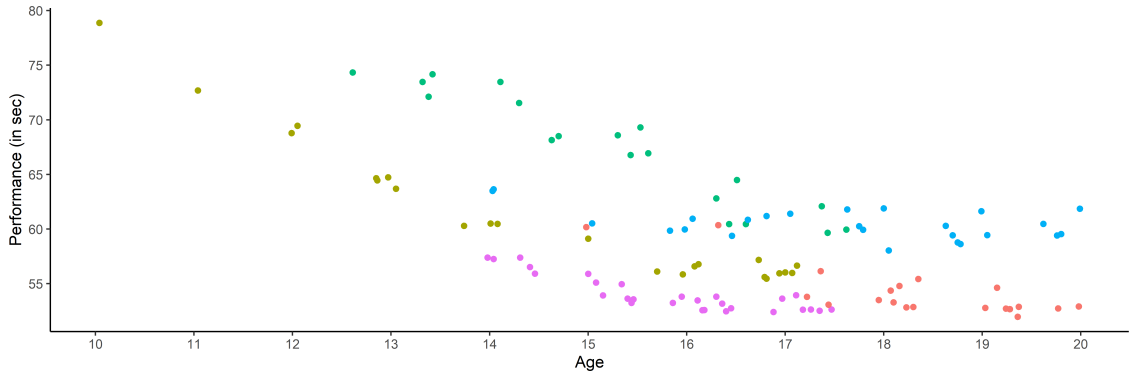

Figure 1: Time series representing the evolution of performances in 100m freestyle competitions between 10 and 20 years for 5 different swimmers, differentiated by colors.

$\{x_k\}_k = \{x_1, \ldots, x_K\}$.

We assume data is collected from M different sources, such that a set of N_i input-output values $\left\{\left(t_i^1, y_i(t_i^1)\right), \ldots, \left(t_i^{N_i}, y_i(t_i^{N_i})\right)\right\}$ constitutes the observations for the i-th individual. Below follows additional convenient notation:

- $\mathbf{t}_i = \{t_i^1, \ldots, t_i^{N_i}\}$, the set of timestamps for the i-th individual,
- $\mathbf{y}_i = y_i(\mathbf{t}_i)$, the vector of outputs for the i-th individual,
- $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, the pooled set of all timestamps among individuals,
- $N = card(\mathbf{t})$, the total number of observed timestamps.

Let us stress that the input values may vary both in number and location among individuals, and we refer as a *common grid* of timestamps to the case where $\mathbf{t}_i = \mathbf{t},\ \forall i \in \mathcal{I}$. Otherwise, we call it an *uncommon grid*. Besides, in order to define a GP mixture model, a latent binary random vector $Z_i = (Z_{i1}, \ldots, Z_{iK})^\intercal$ needs to be associated with each individual, indicating in which cluster it belongs. Namely, if the i-th individual comes from the k-th cluster, then $Z_{ik} = 1$ and 0 otherwise. Moreover, we assume these latent variables to come from the same multinomial distribution: $Z_i \sim \mathcal{M}(1, \boldsymbol{\pi}),\ \forall i \in \mathcal{I}$, with a vector of mixing proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\intercal$ and $\sum_{k=1}^{K} \pi_k = 1$.

2.3 Model and Assumptions

Assuming that the i-th individual belongs to the k-th cluster, we can define its functional expression as the sum of a cluster-specific mean process and an individual-specific centred process:

$$y_i(t) = \mu_k(t) + f_i(t) + \varepsilon_i(t),\ \forall t \in \mathcal{T},$$

where:

- $\mu_k(\cdot) \sim \mathcal{GP}(m_k(\cdot), c_{\gamma_k}(\cdot,\cdot))$ is the common mean process of the k-th cluster,

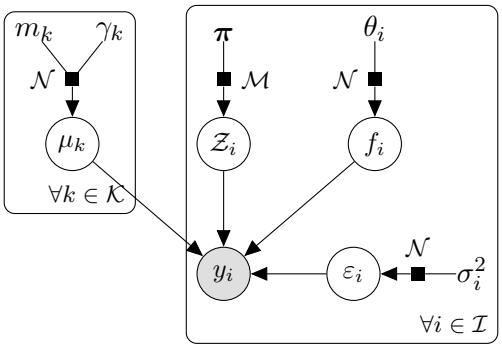

Figure 2: Graphical model of dependencies between variables in the multi-task Gaussian Processes mixture model.

- $f_i(\cdot) \sim \mathcal{GP}(0, \xi_{\theta_i}(\cdot,\cdot))$ is the specific process of the i-th individual,
- $\varepsilon_i(\cdot) \sim \mathcal{GP}(0, \sigma_i^2 I)$ is the error term.

This general model depends upon several mean and covariance parameters, fixed as modelling choices, and hyper-parameters to be estimated:

- $\forall k \in \mathcal{K},\ m_k(\cdot)$ is the prior mean function of the k-th cluster,
- $\forall k \in \mathcal{K},\ c_{\gamma_k}(\cdot,\cdot)$ is the covariance kernel with hyper-parameters γ_k,
- $\forall i \in \mathcal{I},\ \xi_{\theta_i}(\cdot,\cdot)$ is the covariance kernel with hyper-parameters θ_i,
- $\forall i \in \mathcal{I},\ \sigma_i^2 \in \mathbb{R}$ is the noise variance associated with the i-th individual,
- $\forall i \in \mathcal{I},$ we define the shorthand $\psi_{\theta_i,\sigma_i^2}(\cdot,\cdot) = \xi_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I$,
- $\Theta = \left\{ \{\gamma_k\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i, \boldsymbol{\pi} \right\}$, the set of all hyper-parameters of the model.

One can note that we assume here the error term to be individual-specific, although we could also assume it to be cluster-specific and thus indexed by k. Such a choice would result in a valid model since the upcoming developments remain tractable if we substitute ε_k to ε_i everywhere, and associate $\sigma_k^2 I$ with $c_{\gamma_k}(\cdot,\cdot)$ instead of $\xi_{\theta_i}(\cdot,\cdot)$. A discussion about additionally available assumptions on the covariance structures follows in **??**. In this paper, we seek an estimation for Θ among the above quantities, whereas the other objects are pre-specified in the model. For instance, the prior mean $m_k(\cdot)$ is usually set to zero but could also integrate expert knowledge if available. Furthermore, we assume that:

- $\{\mu_k\}_k$ are independent,
- $\{f_i\}_i$ are independent,
- $\{\mathbf{Z}_i\}_i$ are independent,
- $\{\varepsilon_i\}_i$ are independent,
- $\forall i \in \mathcal{I}, \forall k \in \mathcal{K},\ \mu_k,\ f_i,\ \mathbf{Z}_i,\ \varepsilon_i$ and all pairwise combinations are independent.

We display a graphical model on **??** to enlighten the relationships between the different components. From these hypotheses, we can naturally integrate out f_i and derive the conditional prior distribution of $y_i(\cdot)$, providing a hierarchical formulation for the model:

$$y_i(\cdot) \mid \{Z_{ik} = 1, \mu_k(\cdot)\} \sim \mathcal{GP}\left(\mu_k(\cdot), \psi_{\theta_i,\sigma_i^2}(\cdot,\cdot)\right), \ \forall i \in \mathcal{I}, \forall k \in \mathcal{K}.$$

As a consequence, the output processes $\{y_i(\cdot) \mid \{\mathbf{Z}_i\}_i, \{\mu_k(\cdot)\}_k\}_i$ are also independent (conditionally to the latent variables) from one another. Although this model is expressed in terms of infinite-dimensional GPs, we proceed to the inference using finite-dimensional sets of observations $\{\mathbf{t}_i, \mathbf{y}_i\}_i$. Therefore, we can write the joint conditional likelihood of the model (conditioning on the inputs is omitted throughout the paper for clarity):

$$\begin{aligned} p(\{\mathbf{y}_i\}_i \mid \{\mathbf{Z}_i\}_i, \{\mu_k(\mathbf{t})\}_k, \{\theta_i\}_i, \{\sigma_i^2\}_i) &= \prod_{i=1}^{M} p(\mathbf{y}_i \mid \mathbf{Z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i, \sigma_i) \\ &= \prod_{i=1}^{M}\prod_{k=1}^{K} p(\mathbf{y}_i \mid Z_{ik} = 1, \mu_k(\mathbf{t}_i), \theta_i, \sigma_i)^{Z_{ik}} \\ &= \prod_{i=1}^{M}\prod_{k=1}^{K} \mathcal{N}\left(\mathbf{y}_i; \mu_k(\mathbf{t}_i), \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right)^{Z_{ik}}, \end{aligned}$$

where $\forall i \in \mathcal{I}$, $\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i} = \psi_{\theta_i,\sigma_i^2}(\mathbf{t}_i, \mathbf{t}_i) = \left[\psi_{\theta_i,\sigma_i^2}(u,v)\right]_{u,v\in\mathbf{t}_i}$ is a $N_i \times N_i$ covariance matrix. The mean processes being common to all individuals in a cluster, we need to evaluate their prior distributions on the pooled grid of timestamps $\mathbf{t}$:

$$\begin{aligned} p(\{\mu_k(\mathbf{t})\}_k \mid \{\gamma_k\}_k) &= \prod_{k=1}^{K} p(\mu_k(\mathbf{t}) \mid \gamma_k) \\ &= \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right), \end{aligned}$$

where $\mathbf{C}_{\gamma_k}^{\mathbf{t}} = c_{\gamma_k}(\mathbf{t},\mathbf{t}) = [c_{\gamma_k}(k,\ell)]_{k,l\in\mathbf{t}}$ is a $N \times N$ covariance matrix. Finally, the joint distribution of the clustering latent variables also factorises over the individuals:

$$\begin{aligned} p(\{\mathbf{Z}_i\}_i \mid \boldsymbol{\pi}) &= \prod_{i=1}^{M} p(\mathbf{Z}_i \mid \boldsymbol{\pi}) \\ &= \prod_{i=1}^{M} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\pi}) \\ &= \prod_{i=1}^{M}\prod_{k=1}^{K} \pi_k^{Z_{ik}}. \end{aligned}$$

From all these expressions, the complete-data likelihood of the model can be derived:

$$\begin{aligned} p(\{\mathbf{y}_i\}_i, \{\mathbf{Z}_i\}_i, \{\mu_k(\mathbf{t})\}_k \mid \Theta) &= p(\{\mu_k(\mathbf{t})\}_k \mid \gamma_k) \prod_{i=1}^{M} p(\mathbf{y}_i \mid \mathbf{Z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i, \sigma_i^2) p(\mathbf{Z}_i \mid \boldsymbol{\pi}) \\ &= \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right) \prod_{i=1}^{M} \left(\pi_k \mathcal{N}\left(\mathbf{y}_i; \mu_k(\mathbf{t}_i), \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right)\right)^{Z_{ik}}. \end{aligned}$$

This expression would usually serve to estimate the hyper-parameters Θ, although it depends here on latent variables that cannot be evaluated directly. Even if the prior distributions over $\{\mathbf{Z}_i\}_i$ and $\{\mu_k(\mathbf{t})\}_k$ are independent, the expressions of their respective posteriors would inevitably depend on each other. Nevertheless, it remains possible to derive variational approximations for these distributions that still factorise nicely over the terms $\mathbf{Z}_i, \forall i \in \mathcal{I}$, and $\mu_k(\mathbf{t}), \forall k \in \mathcal{K}$. Consequently, the following inference procedure involves a variational EM algorithm that we shall detail after a quick discussion on the optional hypotheses for the model.

2.4 Assumptions on the Covariance Structure

Throughout this paper, we detail a common ground procedure that remains consistent regardless of the covariance structure of the considered GPs. Notice that we chose a parametric distinction of the covariance kernels through the definition of hyper-parameters, different from one individual to another. However, there are no theoretical restrictions on the underlying form of the considered kernels, and we indicate a differentiation on the sole hyper-parameters merely for convenience in writing. A usual kernel in the GP literature is known as the *exponentiated quadratic* kernel (also called sometimes squared exponential or radial basis function kernel). This kernel only depends upon two hyper-parameters $\theta = \{v, \ell\}$ such as:

$$k_{\text{EQ}}\left(x, x'\right) = v^2 \exp\left(-\frac{\left(x - x'\right)^2}{2\ell^2}\right). \tag{1}$$

Although beyond the scope of the present paper, let us mention the existence of a rich literature on kernel design, properties and combinations: see **?**, Chapter 4 or **?** for comprehensive studies.

In the algorithm Magma (**?**), the multi-task aspect is mainly supported by the mean process, although the model also allows information sharing among individuals through the covariance structure. These two aspects being constructed independently, we could think of the model as potentially *double multi-task*, both in mean and covariance. More precisely, if we assume $\{\{\theta_i\}_i, \{\sigma_i^2\}_i\} = \{\theta_0, \sigma_0^2\}$, $\forall i \in \mathcal{I}$, then all f_i are assumed to be different realisations of the same GP, and thus all individuals contributes to the estimation of the common hyper-parameters. Hence, such an assumption that may appear restrictive at first glance actually offers a valuable way to share common patterns between tasks. Furthermore, the same kind of hypothesis can be proposed at the cluster level with $\{\gamma_k\}_k = \gamma_0$, $\forall k \in \mathcal{K}$. In this case, we would assume that all the clusters' mean processes $\{\mu_k\}_k$ share the same covariance structure. This property would indicate that the patterns, or the variations of the curves, are expected to be roughly identical from one cluster to another and that the differentiation would be mainly due to the mean values. Conversely, different covariance structures across kernels offer additional flexibility for the clusters to differ both in position and in trend, smoothness, or any property that could be coded in a kernel. Speaking rather loosely, we may think of these different settings as a trade-off between flexibility and information sharing, or as a choice between an individual or collective modelling of the covariance. Overall, our algorithm provides 4 different settings, offering a rather wide range of assumptions for an adequate adaptation to different applicative situations. Note that the computational considerations are also of paramount importance when it comes to optimising a likelihood over a potentially high number of parameters. Hence, we display on **??** a summary of the 4 different settings, providing

Table 1 Summary of the 4 available assumptions on the hyper-parameters, with their respective shortening notation and the associated number of sets of hyper-parameters (HPs) to optimise.

	$\theta_0 = \theta_i, \forall i \in \mathcal{I}$		$\theta_i \neq \theta_j, \forall i \neq j$	
	Notation	Nb of HPs	Notation	Nb of HPs
$\gamma_0 = \gamma_k, \forall k \in \mathcal{K}$	$\mathcal{H}_{00}$	2	$\mathcal{H}_{0i}$	M + 1
$\gamma_k \neq \gamma_l, \forall k \neq l$	$\mathcal{H}_{k0}$	K + 1	$\mathcal{H}_{ki}$	M + K

a shortening notation along with the associated number of hyper-parameters (or sets of hyper-parameters in the case of θ_i and γ_k) that are required to be learnt in practice.

3 Inference

Although a fully Bayesian point-of-view could be taken on the learning procedure by defining prior distributions of the hyper-parameters and directly use an MCMC algorithm (**??**) for approximate inference on the posteriors, this approach remains computationally challenging in practice. Conversely, variational methods have proved to be highly efficient to conduct inference in difficult GP problems (**??**) and may apply in our context as well. By introducing an adequate independence assumption, we are able to derive a variational formulation leading to analytical approximations for the true hyper-posterior distributions of the latent variables. Then, these hyper-posterior updates allow the computation of a lower bound of the true log-likelihood, thereby specifying the E step of the VEM algorithm (**?**) that conducts the overall inference. Alternatively, we can maximise this lower bound with respect to the hyper-parameters in the M step for optimisation purpose, to provide estimates. By iterating on these two steps until convergence (pseudo-code in **??**), the procedure is proved to reach local optima of the lower bound (**?**), usually in a few iterations. For the sake of clarity, the shorthand $\mathbf{Z} = \{\mathbf{Z}_i\}_i$ and $\boldsymbol{\mu} = \{\mu_k(\mathbf{t})\}_k$ is used in this section when referring to the corresponding set of latent variables.

3.1 Variational EM Algorithm

We seek an appropriate and analytical approximation $q(\mathbf{Z}, \boldsymbol{\mu})$ for the exact hyper-posterior distribution $p(\mathbf{Z}, \boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \Theta)$. Let us first notice that for any distribution $q(\mathbf{Z}, \boldsymbol{\mu})$, the following decomposition holds for the observed-data log-likelihood:

$$\log p(\{\mathbf{y}_i\}_i \mid \Theta) = \mathrm{KL}\left(q \| p\right) + \mathcal{L}(q; \Theta), \tag{2}$$

with:

$$\mathrm{KL}\left(q \| p\right) = \int\int q(\mathbf{Z}, \boldsymbol{\mu}) \log \frac{q(\mathbf{Z}, \boldsymbol{\mu})}{p(\mathbf{Z}, \boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \Theta)} \, \mathrm{d}\mathbf{Z} \, \mathrm{d}\boldsymbol{\mu},$$
$$\mathcal{L}(q; \Theta) = -\int\int q(\mathbf{Z}, \boldsymbol{\mu}) \log \frac{q(\mathbf{Z}, \boldsymbol{\mu})}{p(\mathbf{Z}, \boldsymbol{\mu}, \{\mathbf{y}_i\}_i \mid \Theta)} \, \mathrm{d}\mathbf{Z} \, \mathrm{d}\boldsymbol{\mu}.$$

Therefore, we expressed the intractable log-likelihood of the model by introducing the Kullback-Leibler (KL) divergence between the approximation $q(\mathbf{Z}, \boldsymbol{\mu})$ and the corresponding true distribution $p(\mathbf{Z}, \boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \Theta)$. The right-hand term $\mathcal{L}(q; \Theta)$ in (**??**) defines a so-called *lower bound* for $\log p(\{\mathbf{y}_i\}_i \mid \Theta)$ since a KL divergence is nonnegative by definition. This lower bound depends both upon the approximate distribution $q(\cdot)$ and the

hyper-parameters Θ, while remaining tractable under adequate assumptions. By maximising $\mathcal{L}(q;\Theta)$ alternatively with respect to both quantities, optima for the hyper-parameters shall be reached. To achieve such a procedure, the following factorisation is assumed for the approximated distribution:

$$q(\mathbf{Z},\mu) = q_{\mathbf{Z}}(\mathbf{Z})q_{\boldsymbol{\mu}}(\boldsymbol{\mu}).$$

Colloquially, we could say that the independence property that lacks to compute explicit hyper-posterior distributions is *imposed*. Such a condition restricts the family of distributions from which we choose $q(\cdot)$, and we now seek approximations within this family that are as close as possible to the true hyper-posteriors.

E step

In the expectation step (E step) of the VEM algorithm, the lower bound of the marginal likelihood $\mathcal{L}(q;\Theta)$ is maximised with respect to the distribution $q(\cdot)$, considering that initial or previously estimated values for $\widehat{\Theta}$ are available. Making use of the factorised form previously assumed, we can derive analytical expressions for the optimal distributions over $q_{\mathbf{Z}}(\mathbf{Z})$ and $q_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. As the computing of each distribution involves taking an expectation with respect to the other one, this suggests an iterative procedure where whether the initialisation or a previous estimation serves in the current optimisation process. Therefore, two propositions are introduced below, respectively detailing the exact derivation of the optimal distributions $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$ and $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ (all proofs are deferred to the corresponding **??**).

Proposition 3.1. *Assume that the hyper-parameters $\widehat{\Theta}$ and the variational distribution $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right)$ are known. The optimal variational approximation $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$ of the true hyper-posterior $p\left(\mathbf{Z} \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}\right)$ factorises as a product of multinomial distributions:*

$$\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{iN})^{\intercal}\right), \tag{3}$$

where:

$$\tau_{ik} = \frac{\widehat{\pi}_k \, \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}\right) \exp\left(-\frac{1}{2} tr\left(\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}{}^{-1} \widehat{\mathbf{C}}_k^{\mathbf{t}_i}\right)\right)}{\sum_{l=1}^{K} \widehat{\pi}_l \, \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_l(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}\right) \exp\left(-\frac{1}{2} tr\left(\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}{}^{-1} \widehat{\mathbf{C}}_l^{\mathbf{t}_i}\right)\right)}, \ \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{4}$$

Proposition 3.2. *Assume that the hyper-parameters $\widehat{\Theta}$ and the variational distribution $\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i\right)$ are known. The optimal variational approximation $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ of the true hyper-posterior $p\left(\boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}\right)$ factorises as a product of multivariate Gaussian distributions:*

$$\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right), \tag{5}$$

with:

- $\widehat{\mathbf{C}}_k^{\mathbf{t}} = \left(\mathbf{C}^{\mathbf{t}}_{\widehat{\gamma}_k}{}^{-1} + \sum_{i=1}^{M} \tau_{ik} \tilde{\boldsymbol{\Psi}}_i^{-1}\right)^{-1}, \ \forall k \in \mathcal{K},$

- $\widehat{m}_k(\mathbf{t}) = \widehat{\mathbf{C}}_k^{\mathbf{t}} \left({\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} m_k(\mathbf{t}) + \sum_{i=1}^{M} \tau_{ik} \tilde{\mathbf{\Psi}}_i^{-1} \tilde{\mathbf{y}}_i \right), \ \forall k \in \mathcal{K},$

where the following shorthand notation is used:

- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t) \right)_{t \in \mathbf{t}}$ *(N-dimensional vector),*
- $\tilde{\mathbf{\Psi}}_i = \left[\mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\widehat{\theta}_i, \widehat{\sigma}_i^2}(t,t') \right]_{t,t' \in \mathbf{t}}$ *($N \times N$ matrix).*

Notice that the *forced* factorisation we assumed between $\mathbf{Z}$ and $\boldsymbol{\mu}$ for approximation purpose additionally offers an induced independence between individuals as indicated by the factorisation in (**??**), and between clusters in (**??**).

M step

At this point, we have fixed an estimation for $q(\cdot)$ in the lower bound that shall serve to handle the maximisation of $\mathcal{L}(\widehat{q}, \Theta)$ with respect to the hyper-parameters. This maximisation step (M step) depends on the initial assumptions on the generative model (**??**), resulting in four different versions for the VEM algorithm (the E step is common to all of them, the branching point is here).

Proposition 3.3. *Assume the variational distributions* $\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i)$ *and* $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right)$ *to be known. For a set of hyper-parameters* $\Theta = \{\{\gamma_k\}_k, \{\theta_i\}_i, \{\sigma_i^2\}_i, \boldsymbol{\pi}\}$*, the optimal values are given by:*

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \, \mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}} \left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \Theta) \right],$$

where $\mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}$ *indicates an expectation taken with respect to* $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ *and* $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$*. In particular, optimal values for* $\boldsymbol{\pi}$ *can be computed explicitly with:*

$$\widehat{\pi}_k = \frac{1}{M} \sum_{i=1}^{M} \tau_{ik}, \ \forall k \in \mathcal{K}.$$

The remaining hyper-parameters are estimated by numerically solving the following maximisation problems, according to the situation. Let us note:

$$\mathcal{L}_k(\mathbf{x}; \mathbf{m}, S) = \log \mathcal{N}(\mathbf{x}; \mathbf{m}, S) - \frac{1}{2} tr\left(\widehat{\mathbf{C}}_k^{\mathbf{t}} S^{-1} \right),$$
$$\mathcal{L}_i(\mathbf{x}; \mathbf{m}, S) = \sum_{k=1}^{K} \tau_{ik} \left(\log \mathcal{N}(\mathbf{x}; \mathbf{m}, S) - \frac{1}{2} tr\left(\widehat{\mathbf{C}}_k^{\mathbf{t}_i} S^{-1} \right) \right).$$

Then, for hypothesis $\mathcal{H}_{ki}$*:*

- $\widehat{\gamma}_k = \underset{\gamma_k}{\operatorname{argmax}} \, \mathcal{L}_k\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}} \right), \ \forall k \in \mathcal{K},$
- $(\widehat{\theta}_i, \widehat{\sigma}_i^2) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \, \mathcal{L}_i\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \mathbf{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i} \right), \ \forall i \in \mathcal{I}.$

For hypothesis $\mathcal{H}_{k0}$*:*

- $\widehat{\gamma}_k = \underset{\gamma_k}{\operatorname{argmax}} \, \mathcal{L}_k\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}} \right), \ \forall k \in \mathcal{K},$

- $(\widehat{\theta}_0, \widehat{\sigma}_0^2) = \underset{\theta_0, \sigma_0^2}{\operatorname{argmax}} \sum_{i=1}^{M} \mathcal{L}_i \left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_0, \sigma_0^2} \right)$.

For hypothesis $\mathcal{H}_{0i}$:

- $\widehat{\gamma}_0 = \underset{\gamma_0}{\operatorname{argmax}} \sum_{k=1}^{K} \mathcal{L}_k \left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_0} \right)$,
- $(\widehat{\theta}_i, \widehat{\sigma}_i^2) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \mathcal{L}_i \left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2} \right), \ \forall i \in \mathcal{I}$.

For hypothesis $\mathcal{H}_{00}$:

- $\widehat{\gamma}_0 = \underset{\gamma_0}{\operatorname{argmax}} \sum_{k=1}^{K} \mathcal{L}_k \left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_0} \right)$,
- $(\widehat{\theta}_0, \widehat{\sigma}_0^2) = \underset{\theta_0, \sigma_0^2}{\operatorname{argmax}} \sum_{i=1}^{M} \mathcal{L}_i \left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_0, \sigma_0^2} \right)$.

Let us stress that, for each sub-case, explicit gradients are available for the functions to maximise, facilitating the optimisation process with gradient-based methods (**??**). The current version of our code implements those gradients and makes use of them within the L-BFGS-B algorithm (**??**) devoted to the numerical maximisation. As previously discussed, the hypothesis $\mathcal{H}_{ki}$ necessitates to learn $M+K$ sets of hyper-parameters. However, we notice in **??** that the factorised forms defined as the sum of a Gaussian log-likelihoods and trace terms offer a way to operate the maximisations in parallel on simple functions. Conversely, for the hypothesis $\mathcal{H}_{00}$, only 2 sets of hyper-parameters need to be optimised, namely γ_0, and $\{\theta_0, \sigma_0^2\}$. The small number of functions to maximise is explained by the fact that they are defined as larger sums over all individuals (respectively all clusters). Moreover, this context highlights a multi-task pattern in covariance structures, since each individual (respectively cluster) contributes to the learning of shared hyper-parameters. In practice, $\mathcal{H}_{00}$ is far easier to manage, and we generally reach robust optima in a few iterations. On the contrary, the settings with many hyper-parameters to learn, using mechanically less data for each, may lead more often to computational burden or pathological results. The remaining hypotheses, $\mathcal{H}_{0i}$ and $\mathcal{H}_{k0}$, are somehow middle ground situations between the two extremes and might be used as a compromise according to the problem being dealt with.

3.2 Initialisation

Below some modelling choices are discussed, in particular the initialisation of some quantities involved in the VEM algorithm:

- $\{m_k(\cdot)\}_k$; the mean functions from the hyper-prior distributions of the associated mean processes $\{\mu_k(\cdot)\}_k$. As it may be difficult to pre-specify meaningful values in the absence of external or expert knowledge, these values are often assumed to be 0. However, it remains possible to integrate information in the model by this mean. However, as exhibited in **??**, the influence of $\{m_k(\cdot)\}_k$ in hyper-posterior computations decreases rapidly when M grows in a multi-task framework.
- $\{\gamma_k\}_k$, $\{\theta_i\}_i$ and $\left\{\sigma_i^2\right\}_i$; the kernel hyper-parameters. We already discussed that the form itself of kernels has to be chosen as well, but once set, we would advise

initiating $\{\gamma_k\}_k$ and $\{\theta_i\}_i$ with close and reasonable values whenever possible. As usual in GP models, nearly singular covariance matrices and numerical instability may occur for pathological initialisations, in particular for the hypotheses, like $\mathcal{H}_{ki}$, with many hyper-parameters to learn. This behaviour frequently occurs in the GP framework, and one way to handle this issue is to add a so-called *jitter* term (**?**) on the diagonal of the ill-defined covariance matrices.

- $\{\tau_{ik}\}_{ik}$; the estimated individual membership probabilities (or $\boldsymbol{\pi}$; the prior vector of clusters' proportions). Both quantities are valid initialisation depending on whether we start the VEM iterations by an E step or an M step. If we only want to set the initial proportions of each cluster in the absence of additional information, we may merely specify $\boldsymbol{\pi}$ and start with an E step. Otherwise, if we insert the results from a previous clustering algorithm as an initialisation, the probabilities τ_{ik} for each individual and cluster can be fully specified before proceeding to an M step (or to the $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$'s computing and then the M step).

Let us finally stress that the convergence (to local maxima) of VEM algorithms partly depends on these initialisations. Different strategies have been proposed in the literature to manage this issue, among which simulated annealing (**?**) or repeated short runs (**?**).

3.3 Pseudocode

The overall algorithm is called MagmaClust (as an extension of the algorithm Magma to cluster-specific mean GPs) and we provide below the pseudo-code summarising the inference procedure. The corresponding R code is available at https://github.com/ArthurLeroy/MAGMAclust.

Algorithm 1 MagmaClust: Variational EM algorithm

Initialise $\{m_k(\mathbf{t})\}_k$, $\Theta = \left\{\{\gamma_k\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i\right\}$ and $\{\boldsymbol{\tau}_i^{ini}\}_i$ (or $\boldsymbol{\pi}$).
while not converged **do**
E step: Optimise $\mathcal{L}(q;\Theta)$ w.r.t. $q(\cdot)$:

$$\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i).$$

$$\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}).$$

M step: Optimise $\mathcal{L}(q;\Theta)$ w.r.t. Θ:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\mathbf{Z},\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \Theta)\right].$$

end while
return $\widehat{\Theta}$, $\{\boldsymbol{\tau}_i\}_i$ $\{\widehat{m}_k(\mathbf{t})\}_k$, $\{\widehat{\mathbf{C}}_k^{\mathbf{t}}\}_k$.

3.4 Model Selection

The question of the adequate choice of K in clustering applications is a recurrent concern in practice. Many criteria have been introduced in the literature, among which those relying on penalisation of the likelihood like AIC (**?**) or BIC (**?**) for instance. Whereas we seek a BIC-like formula, we can recall that the likelihood $p(\{\mathbf{y}_i\}_i \mid \widehat{\Theta})$ cannot be computed directly in the present context. However, as for inference, we may still use the previously introduced lower bound $\mathcal{L}(\widehat{q}; \widehat{\Theta})$ to adapt a so-called variational-BIC (VBIC) quantity to

maximise, as proposed in **?**. The expression of this criterion is provided below while we defer the full derivation of the lower bound to **??**.

Proposition 3.4. *After convergence of the VEM algorithm, a variational-BIC expression can be derived as:*

$$\begin{aligned} BIC_{var}(K) &= \mathcal{L}(\widehat{q}; \widehat{\Theta}) - \frac{\text{card}\{HP\}}{2} \log M \\ &= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\tau_{ik} \left(\log \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2} \right) - \frac{1}{2} tr\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k \mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}{}^{-1} \right) + \log \frac{\widehat{\pi_k}}{\tau_{ik}} \right) \right] \\ &\quad + \sum_{k=1}^{K} \left[\log \mathcal{N}\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\widehat{\gamma}_k} \right) - \frac{1}{2} tr\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k \mathbf{C}^{\mathbf{t}}_{\widehat{\gamma}_k}{}^{-1} \right) \right. \\ &\qquad \left. + \frac{1}{2} \log \left| \widehat{\mathbf{C}}^{\mathbf{t}}_k \right| + N \log 2\pi + N \right] - \frac{\alpha_i + \alpha_k + (K-1)}{2} \log M, \end{aligned}$$

where:

- α_i *is the number of hyper-parameters from the individual processes' kernels,*
- α_k *is the number of hyper-parameters from the mean processes' kernels,*
- $K-1$ *is the number of free parameters* $\widehat{\pi_k}$ *(because of the constraint* $\sum_{k=1}^{K} \pi_k = 1$ *).*

Let us mention that the numbers α_i and α_k in the penalty term vary according to the considered modelling hypothesis ($\mathcal{H}_{00}$, $\mathcal{H}_{k0}$, $\mathcal{H}_{0i}$ or $\mathcal{H}_{ki}$), see **??** for details.

4 Prediction

At this point, we would consider that the inference on the model is completed, since the training data set of observed individuals $\{\mathbf{y}_i\}_i$ enabled to estimate the desired hyper-parameters and the distributions of latent variables. For the sake of concision, we thus omit the writing of conditionings over $\widehat{\Theta}$ in the sequel. Recalling our illustrative example, we would have used competition's results over a long period from thousands of swimmers for training the model, and we now expect to make predictions of future performances for any young athlete in the early stages of their career. Therefore, let us assume the partial observation of a new individual, denoted by the index $*$, for whom we collected a few data points $y_*(\mathbf{t}_*)$ at timestamps $\mathbf{t}_*$. Defining a multi-task GPs mixture prediction consists in seeking an analytical distribution $p(y_*(\cdot) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$, according to the information brought by: its own observations; the training data set; the cluster structure among individuals. As we aim at studying the output values $y_*(\cdot)$ at arbitrarily chosen timestamps, say $\mathbf{t}^p$ (the index p stands for *prediction*), we propose a new notation for the pooled vector of timestamps $\mathbf{t}^p_* = \begin{bmatrix} \mathbf{t}^p \\ \mathbf{t}_* \end{bmatrix}$. This vector serves as a working grid on which the different distributions involved in the prediction procedure are evaluated. In the absence of external restrictions, we would strongly advise to include the observed timestamps of all training individuals, $\mathbf{t}$, within $\mathbf{t}^p_*$, since evaluating the processes at these locations allows for sharing information across tasks. Otherwise, any data points defined on timestamps outside of the working grid would be discarded from the multi-task aspect of the model.

Table 2 Summary of the different steps to perform in the prediction procedure, according to the model assumptions and the target grid of timestamps.

	$\mathbf{t}_*^p = \mathbf{t}$	$\mathbf{t}_*^p \neq \mathbf{t}$
$\mathcal{H}_{00}$	2-3bis-4-5	1-2-3bis-4-5
$\mathcal{H}_{k0}$	2-3bis-4-5	1-2-3bis-4-5
$\mathcal{H}_{0i}$	2-3-4-5	1-2-3-4-5
$\mathcal{H}_{ki}$	2-3-4-5	1-2-3-4-5

In particular, if $\mathbf{t}_*^p = \mathbf{t}$, we may even use directly the variational distribution $q_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ computed in the VEM algorithm, and thus skip one step of the prediction procedure that is described below. Throughout the section, we aim at defining a probabilistic prediction for this new individual, accounting for the information of all training data $\{\mathbf{y}_i\}_i$. To this end, we manipulate several distributions of the type $p(\cdot \mid \{\mathbf{y}_i\}_i)$ and refer to them with the adjective *multi-task*. Additionally to highlighting the information-sharing aspect across individuals, this term allows us to distinguish the role of $\{\mathbf{y}_i\}_i$ from the one of the newly observed data $y_*(\mathbf{t}_*)$, which are now the reference data for establishing if a distribution is called a *prior* or a *posterior*. Deriving a predictive distribution in our multi-task GP framework requires to complete the following steps.

1. Compute the hyper-posterior approximation of $\{\mu_k(\cdot)\}_k$ at $\mathbf{t}_*^p$: $\widehat{q}_\mu(\{\mu_k(\mathbf{t}_*^p)\}_k)$,
2. Deduce the multi-task prior distribution: $p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i)$,
3. Compute the new hyper-parameters $\{\theta_*, \sigma_*^2\}$ and $p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$ via an EM,

3bis. Assign $\theta_* = \theta_0$, $\sigma_*^2 = \sigma_0^2$ and compute directly $p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$,

4. Compute the multi-task posterior distribution: $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \mathbf{Z}_*, \{\mathbf{y}_i\}_i)$,
5. Deduce the multi-task GPs mixture prediction: $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$.

We already discussed the influence of the initial modelling hypotheses on the overall procedure. Hence, let us display in **??** a quick reminder helping to keep track of which steps need to be performed in each context.

4.1 Posterior Inference on the Mean Processes

In order to integrate the information contained in the shared mean processes, we first need to re-compute the variational approximation of $\{\mu_k(\cdot)\}_k$'s hyper-posterior on the new $\tilde{N}$-dimensional working grid $\mathbf{t}_*^p$. By using once more **??**, it appears straightforward to derive this quantity that still factorises as a product of Gaussian distributions where we merely substitute the values of timestamps:

$$\widehat{q}_{\boldsymbol{\mu}}(\{\mu_k(\mathbf{t}_*^p)\}_k) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}_*^p); \widehat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p}\right),$$

with:

- $\widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} = \left(\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}_*^p\,-1} + \sum_{i=1}^{M} \tau_{ik} \tilde{\boldsymbol{\Psi}}_i^{-1}\right)^{-1}, \ \forall k \in \mathcal{K},$

- $\widehat{m}_k(\mathbf{t}_*^p) = \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p}\left(\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}_*^p\,-1} m_k(\mathbf{t}_*^p) + \sum_{i=1}^{M} \boldsymbol{\tau}_{ik}\tilde{\boldsymbol{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i\right),\ \forall k \in \mathcal{K},$

where the following shorthand notation is used:

- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t\in\mathbf{t}_i]} \times y_i(t)\right)_{t\in\mathbf{t}_*^p}$ ($\tilde{N}$-dimensional vector),
- $\tilde{\boldsymbol{\Psi}}_i = \left[\mathbb{1}_{[t,t'\in\mathbf{t}_i]} \times \psi_{\widehat{\theta}_i,\widehat{\sigma}_i^2}(t,t')\right]_{t,t'\in\mathbf{t}_*^p}$ ($\tilde{N}\times\tilde{N}$ matrix).

We acknowledge that the subsequent analytical developments party rely on this variational approximate distribution $\widehat{q}_{\boldsymbol{\mu}}(\{\mu_k(\mathbf{t}_*^p)\}_k)$, and may thus be considered, in a sense, as approximated as well. However, this quantity provides a valuable closed-form expression that we can substitute to the true hyper-posterior in **??** below.

4.2 Computation of the Multi-Task Prior Distributions

For a sake of completeness, let us recall the equivalence between two ways of writing conditional distributions that are used in the subsequent results:

$$p(\cdot \mid \mathbf{Z}_*) = \prod_{k=1}^{K} p(\cdot \mid Z_{*k} = 1)^{Z_{*k}}.$$

We may regularly substitute one to the other in the sequel depending on the handier in the context. Once the mean processes' distributions are re-computed on the working grid, their underlying influence shall be directly plugged into a marginalised multi-task prior over $y_*(\mathbf{t}_*^p)$ by integrating out the $\{\mu_k(\mathbf{t}_*^p)\}_k$. As the mean processes vanish, the new individual's outputs $y_*(\mathbf{t}_*^p)$ directly depends upon the training data set $\{\mathbf{y}_i\}_i$, as highlighted in the proposition below.

Proposition 4.1. *For a set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of y_* knowing its clustering latent variable is given by:*

$$p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i) = \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*^p); \widehat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*^p}\right)^{Z_{*k}}. \tag{6}$$

Proof. Let us recall that, conditionally to their mean process, the individuals are independent of one another. Then, for all $k \in \mathcal{K}$, we have:

$$\begin{aligned}
p(y_*(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i) &= \int p(y_*(\mathbf{t}_*^p), \mu_k(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i)\, \mathrm{d}\mu_k(\mathbf{t}_*^p) \\
&= \int p(y_*(\mathbf{t}_*^p) \mid \mu_k(\mathbf{t}_*^p), Z_{*k} = 1)\, \underbrace{p(\mu_k(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i)}_{\approx q_{\boldsymbol{\mu}}(\mu_k(\mathbf{t}_*^p))}\, \mathrm{d}\mu_k(\mathbf{t}_*^p) \\
&\approx \int \mathcal{N}\left(y_*(\mathbf{t}_*^p); \mu_k(\mathbf{t}_*^p), \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*^p}\right) \mathcal{N}\left(\mu_k(\mathbf{t}_*^p); \widehat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p}\right) \mathrm{d}\mu_k(\mathbf{t}_*^p) \\
&= \mathcal{N}\left(y_*(\mathbf{t}_*^p); \widehat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*^p}\right).
\end{aligned}$$

The final line is obtained by remarking that such a convolution of Gaussian distributions remains Gaussian as well (**?**, Chapter 2), and we refer to **?** for the detailed calculus in this exact context. Therefore, we finally get:

$$p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i) = \prod_{k=1}^{K} p(y_*(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i)^{Z_{*k}}$$
$$= \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*^p); \widehat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*^p}\right)^{Z_{*k}}.$$

□

4.3 Optimisation of the New Hyper-Parameters and Computation of the Clusters' Probabilities

Now that the mean processes have been removed at the previous step, this section strongly resembles the classical learning procedure through an EM algorithm for a Gaussian mixture model. In our case, it allows us both to estimate the hyper-parameters of the new individual $\{\theta_*, \sigma_*\}$ and to compute the hyper-posterior distribution of its latent clustering variable $\mathbf{Z}_*$, which provides the associated clusters' membership probabilities $\boldsymbol{\tau}_*$. As before, E steps and M steps are alternatively processed until convergence, but this time by working with exact formulations instead of variational ones.

E step

In the E step, hyper-parameters estimates are assumed to be known. Recalling that the latent clustering variable $\mathbf{Z}_*$ is independent from the training data $\{\mathbf{y}_i\}_i$, the multi-task hyper-posterior distribution maintains an explicit derivation:

$$p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i, \widehat{\theta}_*, \widehat{\sigma}_*^2, \widehat{\boldsymbol{\pi}}) \propto p(y_*(\mathbf{t}_*) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i, \widehat{\theta}_*, \widehat{\sigma}_*^2) p(\mathbf{Z}_* \mid \widehat{\boldsymbol{\pi}})$$
$$\propto \prod_{k=1}^{K} \left\{ \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\widehat{\theta}_*,\widehat{\sigma}_*^2}^{\mathbf{t}_*}\right)^{Z_{*k}} \right\} \prod_{l=1}^{K} \widehat{\pi}_l^{Z_{*l}}$$
$$\propto \prod_{k=1}^{K} \left(\widehat{\pi}_k \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\widehat{\theta}_*,\widehat{\sigma}_*^2}^{\mathbf{t}_*}\right)\right)^{Z_{*k}}.$$

By inspection, we recognise the form of a multinomial distribution and thus retrieve the corresponding normalisation constant to deduce:

$$p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i, \widehat{\theta}_*, \widehat{\sigma}_*^2, \widehat{\boldsymbol{\pi}}) = \mathcal{M}\left(\mathbf{Z}_*; 1, \boldsymbol{\tau}_* = (\tau_{*1}, \ldots, \tau_{*K})^\intercal\right), \tag{7}$$

with:

$$\tau_{*k} = \frac{\widehat{\pi}_k \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\widehat{\theta}_*,\widehat{\sigma}_*^2}^{\mathbf{t}_*}\right)}{\sum_{l=1}^{K} \widehat{\pi}_l \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_l(\mathbf{t}_*), \widehat{\mathbf{C}}_l^{\mathbf{t}_*} + \Psi_{\widehat{\theta}_*,\widehat{\sigma}_*^2}^{\mathbf{t}_*}\right)}, \ \forall k \in \mathcal{K}. \tag{8}$$

M step

Assuming to know the value of $\boldsymbol{\tau}_*$, we may derive optimal values for the hyper-parameters of the new individual through the following maximisation:

$$\{\widehat{\theta}_*, \widehat{\sigma}_*^2\} = \underset{\theta_*,\sigma_*}{\operatorname{argmax}} \, \mathbb{E}_{\mathbf{Z}_*}\left[\log p(y_*(\mathbf{t}_*), \mathbf{Z}_* \mid \{\mathbf{y}_i\}_i, \theta_*, \sigma_*, \widehat{\boldsymbol{\pi}})\right].$$

Let us note $\mathcal{L}_*(\theta_*, \sigma_*) = \log p(y_*(\mathbf{t}_*), \mathbf{Z}_* \mid \{\mathbf{y}_i\}_i, \theta_*, \sigma_*, \widehat{\boldsymbol{\pi}})$. By remarking that $\widehat{\boldsymbol{\pi}}$ has already been estimated previously, we may easily derive the expression to maximise with respect to θ_* and σ_* in practice:

$$\begin{aligned}
\mathbb{E}_{\mathbf{Z}_*}\left[\,\mathcal{L}_*(\theta_*, \sigma_*)\,\right] &= \mathbb{E}_{\mathbf{Z}_*}\left[\log p(y_*(\mathbf{t}_*), \mathbf{Z}_* \mid \{\mathbf{y}_i\}_i, \theta_*, \sigma_*, \widehat{\boldsymbol{\pi}}\,]\right) \\
&= \mathbb{E}_{\mathbf{Z}_*}\left[\log p(y_*(\mathbf{t}_*) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i, \theta_*, \sigma_*) + \log p(\mathbf{Z}_* \mid \widehat{\boldsymbol{\pi}})\,\right] \\
&= \mathbb{E}_{\mathbf{Z}_*}\left[\log \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*}\right)^{Z_{*k}}\right] + C_1 \\
&= \sum_{k=1}^{K} \mathbb{E}_{\mathbf{Z}_*}\left[\,Z_{*k}\,\right] \log \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*}\right) + C_1 \\
&= \sum_{k=1}^{K} \tau_{*k} \log \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*}\right) + C_1,
\end{aligned}$$

where C_1 is a constant term. Thus, the optimisation in this case merely relies on the maximisation of a weighted sum of Gaussian log-likelihoods, for which gradients are well-known.

3bis.

In the case where the hyper-parameters are supposed to be common across individuals ($\mathcal{H}_{00}$ or $\mathcal{H}_{k0}$), there is no need to additional optimisation since we already have $\widehat{\theta}_* = \widehat{\theta}_0$ and $\widehat{\sigma}_*^2 = \widehat{\sigma}_0$ by definition. However, the probabilities of lying in each cluster $\boldsymbol{\tau}_*$ for the new individual still need to be computed, which shall be handled by directly using the expression (**??**) from the E step.

3ter.

Conversely, even if hyper-parameters for each individual are supposed to be different ($\mathcal{H}_{0i}$ or $\mathcal{H}_{ki}$), it remains possible to avoid the implementation of an EM algorithm by stating $\boldsymbol{\tau}_* = \widehat{\boldsymbol{\pi}}$. Such an assumption intuitively expresses that we would guess the membership probabilities of each cluster from the previously estimated mixing proportions, without taking new individual's observations into account. Although we would not recommend this choice for getting optimal results, it still seems to be worth a mention for applications with a compelling need to avoid EM's extra computations during the prediction process.

4.4 Computation of the Multi-Task Posterior Distributions

Once the needed hyper-parameters have been estimated and the prior distribution established, the classical formula for GP predictions can be applied to the new individual, for each possible latent cluster. First, let us recall the prior distribution by separating observed from target timestamps, and introducing a shorthand notation for the covariance:

$$p(y_*(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i) = \mathcal{N}\left(\begin{bmatrix} y_*(\mathbf{t}^p) \\ y_*(\mathbf{t}_*) \end{bmatrix}; \begin{bmatrix} \widehat{m}_k(\mathbf{t}^p) \\ \widehat{m}_k(\mathbf{t}_*) \end{bmatrix}, \begin{pmatrix} \boldsymbol{\Gamma}_k^{\mathbf{t}^p\mathbf{t}^p} & \boldsymbol{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*} \\ \boldsymbol{\Gamma}_k^{\mathbf{t}_*\mathbf{t}^p} & \boldsymbol{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*} \end{pmatrix}\right), \ \forall k \in \mathcal{K},$$

where $\boldsymbol{\Gamma}_k^{\mathbf{t}^p,\mathbf{t}^p} = \widehat{\mathbf{C}}_k^{\mathbf{t}^p} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}^p}$ and likewise for the other blocks of the matrices. Therefore, recalling that conditioning on the sub-vector of observed values $y_*(\mathbf{t}_*)$ maintains a Gaussian distribution (**??**), we can derive the multi-task posterior distribution for each latent cluster:

$$p(y_*(\mathbf{t}^p) \mid Z_{*k} = 1, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \mathcal{N}\left(y_*(\mathbf{t}^p); \widehat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right), \ \forall k \in \mathcal{K}, \tag{9}$$

where:

- $\widehat{\mu}_{*k}(\mathbf{t}^p) = \widehat{m}_k(\mathbf{t}^p) + \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*}{\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*}}^{-1}\left(y_*(\mathbf{t}_*) - \widehat{m}_k(\mathbf{t}_*)\right), \ \forall k \in \mathcal{K}$,
- $\widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p} = \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}^p} - \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*}{\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*}}^{-1}\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}^p}, \ \forall k \in \mathcal{K}$.

4.5 Computation of the Multi-Task GPs Mixture Prediction

To conclude, by summing over all possible combinations for the latent clustering variable $\mathbf{Z}_*$, we can derive the final predictive distribution.

Proposition 4.2. *The multi-task GPs mixture posterior distribution for $y_*(\mathbf{t}^p)$ has the following form:*

$$p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \sum_{k=1}^{K} \tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \widehat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right).$$

Proof.
Taking advantage of (**??**) and the multi-task hyper-posterior distribution of $\mathbf{Z}_*$ as computed in (**??**), it is straightforward to integrate out the latent clustering variable:

$$\begin{aligned}
p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) &= \sum_{\mathbf{Z}_*} p(y_*(\mathbf{t}^p), \mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) \\
&= \sum_{\mathbf{Z}_*} p(y_*(\mathbf{t}^p) \mid \mathbf{Z}_*, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) \\
&= \sum_{\mathbf{Z}_*} \prod_{k=1}^{K} \left(\tau_{*k} \ p(y_*(\mathbf{t}^p) \mid Z_{*k} = 1, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)\right)^{Z_{*k}} \\
&= \sum_{\mathbf{Z}_*} \prod_{k=1}^{K} \left(\tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \widehat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right)\right)^{Z_{*k}} \\
&= \sum_{k=1}^{K} \tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \widehat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right),
\end{aligned}$$

where we recall for the transition to the last line that $Z_{*k} = 1$ if the $*$-th individual belongs to the k-th cluster and $Z_{*k} = 0$ otherwise. Hence, summing a product with only one non-zero exponent over all possible combination for $\mathbf{Z}_*$ is equivalent to merely sum over the values of k, and the variable Z_{*k} simply vanishes. □

Alternative predictions

Even though **??** provides an elegant probabilistic prediction in terms of GPs mixture, it remains important to notice that this quantity is no longer a Gaussian distribution. In particular, the distribution of an output value at any point-wise evaluation is expected to differ significantly from a classical Gaussian variable, by being multi-modal for instance. This property is especially true for individuals with high uncertainty about the clusters they probably belong to, whereas the distribution would be close to the a when $\tau_{*k} \approx 1$

for one cluster and almost zero for the others. While we believe that such a GPs mixture distribution highlights the uncertainty resulting from a possible cluster structure in data and offers a rather original view on the matter of GP predictions, some applications may suffer from this non-Gaussian final distribution. Fortunately, it remains pretty straightforward to proceed to a simplification of the clustering inference by assuming that the $*$-individual only belongs to its more probable cluster, which is equivalent to postulate $max\{\tau_{*k}\}_k = 1$ and the others to be zero. In this case, the final Gaussian mixture turns back into a Gaussian distribution, and we retrieve a uni-modal prediction, easily displayed by its mean along with credible intervals.

5 Complexity Analysis for Training and Prediction

It is customary to stress that computational complexity is of paramount importance in GP models as a consequence of their usual cubic (resp. quadratic) cost in the number of data points for learning (resp. prediction). In the case of MagmaClust, we use information from M individuals scattered into K clusters, each of them providing N_i observations, and those quantities mainly specify the overall complexity of the algorithm. Moreover, N refers to the number of distinct timestamps (i.e. $N \leq \sum_{i=1}^{M} N_i$) in the training data set and corresponds to the dimension of the objects involved in the kernel-specific mean processes computations. Typically, the learning complexity would be proportional to one iteration of the VEM algorithm, which requires $\mathcal{O}\left(M \times N_i^3 + K \times N^3\right)$ operations.

Let us stress that this complexity is linear in the number of tasks, which is significantly lower than classical multi-output GP approaches in the literature. As detailed in **?**, algorithms that can be formulated as linear models of coregionalisations typically present a $\mathcal{O}(M^3N^3)$ training complexity when not resorting to sparse approximations. Several approximations have been developed to reduce this cost, for instance, by using pseudo-inputs to decrease the N^3 term, while lowering the M^3 term generally requires underlying independence assumptions **?**. In the *Experiments* section, the computational advantage of MagmaClust is also empirically highlighted in **??**.

From a practical point of view, the hypotheses formulated on the hyper-parameters may influence the constant of this complexity but generally not in more than an order of magnitude. For instance, the models under the assumption $\mathcal{H}_{00}$ usually require less optimisation time in practice, although it does not change the number or the dimensions of the covariance matrices to inverse, which mainly control the overall computing time. The dominating terms in this expression depend on the context, regarding the relative values of M, N_i, N and K. In contexts where the number of individuals M dominates, like with small common grids of timestamps for instance, the left-hand term would control the complexity, and clustering's additional cost would be negligible. Conversely, for a relatively low number of individuals or a large size N for the pooled grid of timestamps, the right-hand term becomes the primary burden, and the computing time increases proportionally to the number of clusters compared to the original Magma algorithm.

During the prediction step, the re-computation of $\{\mu_k(\cdot)\}_k$'s variational distributions implies K inversions of covariance matrices with dimensions depending on the size of the prediction grid $\mathbf{t}_*^p$. In practice though, if we fix a fine grid of target timestamps in advance, this operation can be assimilated to the learning step. In this case, the prediction complexity remains at most in the same order as the usual learning for a single-task GP,

that is $\mathcal{O}(K \times N_*^3)$ (this corresponds to the estimation of the new individual's hyper-parameters, and would decrease to $\mathcal{O}(K \times N_*^2)$ for $\mathcal{H}_{k0}$ or $\mathcal{H}_{00}$). We shall mention that the definition of a fine grid is generally desirable only for low-dimensional applications since we may quickly reach running time or memory limits as the input's dimension grows. In many contexts, most of the time-consuming learning steps can be performed in advance, and the immediate prediction cost for each new individual is negligible in comparison (generally comparable to a single-task GP prediction).

6 Experiments

The present section is dedicated to the evaluation of MagmaClust on both synthetic and real data sets. The performance of the algorithm is assessed in regards to its clustering and forecast abilities. To this purpose, we introduce below the simulation scheme generating the synthetic data along with the measures used to compare our method to alternatives quantitatively. Throughout, the *exponentiated quadratic* (EQ) kernel, as defined in **??**, serves as covariance structure for both generating data and modelling. The manipulation of more sophisticated kernels remains a topic beyond the scope of the present paper, and the EQ proposes a fair common ground for comparison between methods. Thereby, each kernel introduced in the sequel is associated with two hyper-parameters. Namely, $v \in \mathbb{R}^+$ represents a variance term whereas $\ell \in \mathbb{R}^+$ specifies the length-scale. The synthetic data sets are generated following the general procedure below, with minor modifications according to the model assumptions $\mathcal{H}_{00}$, $\mathcal{H}_{k0}$, $\mathcal{H}_{0i}$ or $\mathcal{H}_{ki}$:

1. Define a random working grid $\mathbf{t} \subset [\,0, 10\,]$ of $N = 200$ timestamps to study $M = 50$ individuals, scattered into K clusters,
2. Draw the prior mean functions for $\{\mu_k(\cdot)\}_k$: $m_k(t) = at + b,\ \forall t \in \mathbf{t}, \forall k \in \mathcal{K}$, where $a \in [\,-2, 2\,]$ and $b \in [\,20, 30\,]$,
3. Draw uniformly hyper-parameters for $\{\mu_k(\cdot)\}_k$'s kernels : $\gamma_k = \{v_{\gamma_k}, \ell_{\gamma_k}\},\ \forall k \in \mathcal{K}$, where $v_{\gamma_k} \in \left[\,1, e^3\,\right]$ and $\ell_{\gamma_k} \in \left[\,1, e^1\,\right]$, (or $\gamma_0 = \{v_{\gamma_0}, \ell_{\gamma_0}\}$),
4. Draw $\mu_k(\mathbf{t}) \sim \mathcal{N}\left(m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right), \forall k \in \mathcal{K}$,
5. For all $i \in \mathcal{I}$, draw uniformly the hyper-parameters for individual kernels $\theta_i = \{v_{\theta_i}, \ell_{\theta_i}\}$, where $v_{\theta_i} \in \left[\,1, e^3\,\right]$, $\ell_{\theta_i} \in \left[\,1, e^1\,\right]$, and $\sigma_i^2 \in [\,0, 0.1\,]$, (or $\theta_0 = \{v_{\theta_0}, \ell_{\theta_0}\}$ and σ_0^2),
6. Define $\boldsymbol{\pi} = (\frac{1}{K}, \ldots, \frac{1}{K})^\intercal$ and draw $\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\pi}),\ \forall i \in \mathcal{I}$,
7. For all $i \in \mathcal{I}$ and $Z_{ik} = 1$, draw uniformly a random subset $\mathbf{t}_i \subset \mathbf{t}$ of $N_i = 30$ timestamps, and draw $\mathbf{y}_i \sim \mathcal{N}\left(\mu_k(\mathbf{t}_i), \boldsymbol{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right)$.

This procedure offers data sets for both the individuals $\{\mathbf{t}_i, \mathbf{y}_i\}_i$ and the underlying mean processes $\{\mathbf{t}, \mu_k(\mathbf{t})\}_k$. In the context of prediction, a new individual is generated according to the same scheme, although its first 20 data points are assumed to be observed while the remaining 10 serve as testing values. While it may be argued that this repartition 20-10 is somehow arbitrary, a more detailed analysis with changing numbers of observed points in **?** revealed a low effect on the global evaluation. Unless otherwise stated, we fix the number of clusters to be $K^* = 3$ and the model assumption to be $\mathcal{H}_{00}$ for generating the data. Let us recall that we provided a variational-BIC formula in **??** to

select an appropriate number of clusters K from data. Therefore, this measure is evaluated in following experiments and used for model selection purposes in the real-life application.

Besides, the adjusted rand index (ARI) (**?**) is used as a measure of adequacy for comparison between the groups obtained through the clustering procedure and the true clusters that generated the data. More specifically, the ARI is defined by counting the proportions of matching pairs between groups, and a value of 1 represents a perfect correspondence. One can note that the ARI still applies when it comes to evaluating clustering partitions with different numbers of clusters. On the matter of prediction, the mean square error (MSE) between predicted means and the true values offers a measure of the average forecast performance. Formally, we define the MSE in prediction on the 10 testing points for the new individual as:

$$\frac{1}{10}\sum_{u=21}^{30}\left(y_*^{pred}(t_*^u) - y_*^{true}(t_*^u)\right)^2.$$

Moreover, an additional measure accounting for the validity of uncertainty quantification is defined in **?** as the percentage of true data effectively lying within the 95% credible interval (CI_{95}), which is constructed from the predictive distribution. We extend here this measure to the context of GPs mixture, where CI_{95} is no longer available directly (as for any multi-modal distribution). Namely, the weighted CI_{95} coverage ($WCIC_{95}$) is defined to be:

$$100 \times \frac{1}{10}\sum_{u=21}^{30}\sum_{k=1}^{K}\tau_{*k}\ \mathbb{1}_{\{y_*^{true}(t_*^u)\in\ CI_{95}^k\}},$$

where CI_{95}^k represents the CI_{95} computed for the k-th cluster-specific Gaussian predictive distribution (**??**). In the case where $K = 1$, i.e. a simple Gaussian instead of a GPs mixture, the $WCIC_{95}$ reduces to the previously evoked CI_{95} coverage. By averaging the weighted cluster-specific CI_{95}^k coverage, we still obtain an adequate and comparable quantification of the uncertainty relevance for our predictions. By definition, the value of this indicator should be as close as possible to 95%. Finally, the mean functions $\{m_k(\cdot)\}_k$ are set to be 0 in MagmaClust, as usual for GPs, whereas the membership probabilities τ_{ik} are initialised thanks to a preliminary k-means algorithm.

6.1 Illustration on Synthetic Examples

?? provides a comparison on the same data set between a classical GP regression (top), the multi-task GP algorithm Magma (middle), and the multi-task GPs mixture approach MagmaClust (bottom). On each sub-graph, the plain blue line represents the mean parameter from the predictive distribution, and the grey shaded area covers the CI_{95}. The dashed lines stand for the multi-task prior mean functions $\{\widehat{m}_k(\cdot)\}_k$ resulting from the estimation of the mean processes. The points in black are the observations for the new individual $*$, whereas the red points constitute the true target values to forecast. Moreover, the colourful background points depict the data of the training individuals, which we colour according to their true cluster in MagmaClust displays (bottom). As expected, a simple GP regression provides an adequate fit close to the data points before quickly diving to the prior value 0 when lacking information. Conversely, Magma takes advantage of its multi-task component to share knowledge across individuals by estimating a more relevant mean process. However, this unique mean process appears unable to account for the clear group structure, although adequately recovering the dispersion of the data. In the case of MagmaClust, we display the cluster-specific prediction (**??**) for the most probable group

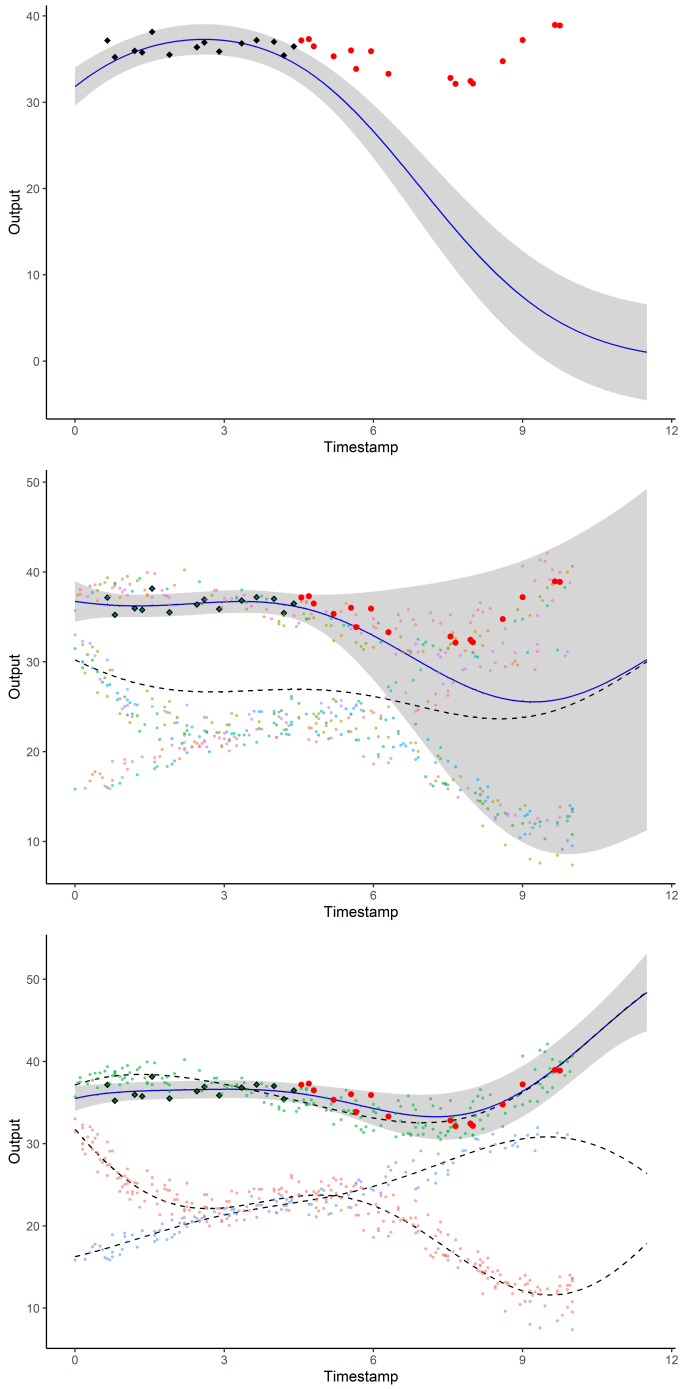

Figure 3: Prediction curves (blue) with associated 95% credible intervals (grey) from GP regression (top), MAGMA (middle) and MAGMACLUST (bottom). The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training data set, coloured relatively to individuals (middle) or clusters (bottom).

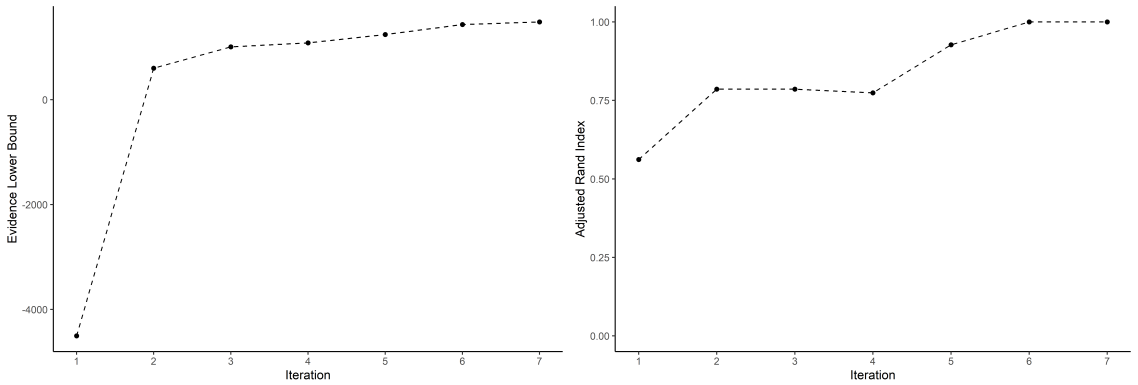

Figure 4: **Left:** Values of the evidence lower bound (ELBO) for successive iterations of the VEM algorithm during MagmaClust training. **Right:** The corresponding ARI values between predicted and true clusters for the same iterations. In this illustrative example, the algorithm reached convergence after 7 iterations only.

instead of the GPs mixture prediction, since $\max_k(\boldsymbol{\tau}_*) \approx 1$ in this example. The model selection method based on maximum VBIC values correctly retrieved the true number of cluster $K = 3$. To illustrate the training procedure dynamics in this example, we provide on **??** a tracking, for each iteration of the VEM algorithm until convergence, of the evidence lower bound (ELBO) and the corresponding ARI between predicted and true clusters. As the ELBO increases, notice that the ARI also improves to finally reach 1 at convergence, which means we fully retrieved the true clusters at the end of training. Although this simple example only depicts a single-run instance, it provides an accurate intuition on the general behaviour of MagmaClust. In practice, the algorithm quickly improves the clustering structure and associated mean processes during the first two steps, and generally converges in a handful of iterations.

Overall, this illustrative example highlights the benefit we can get from considering group-structured similarities between individuals in GP predictions. Notice that our method offers both a significant improvement in mean prediction and a narrowed uncertainty around this value. Additionally, we display on **??** the specific predictions according to the two remaining clusters (although associated with nearly 0 probabilities). We remark that the predictions move towards the cluster specific mean processes as soon as the observations become too distant. In this idealised example, we displayed Gaussian predictive distributions for convenience though, in general, a Gaussian mixture might rarely be unimodal. Therefore, we propose in **??** another example with a higher variance and overlapping groups, where the VBIC still provides the correct number of clusters. While the ARI between predicted and true clusters was equal to 1 (perfect match) in the previous example, it now decreases to 0.78. Moreover, the vector of membership probabilities associated with the **??** for the predicted individual happens to be: $\boldsymbol{\tau}_* = (0.95, 0.05, 0)$. The left-hand graph provides an illustration of the predictive mean, acquired from the multi-task GPs mixture distribution described in **??**. We may notice that this curve lies very close to one cluster's mean although not completely overlapping it, because of the $\tau_{*k} = 0.05$ probability for another cluster, which slightly pulls the prediction onto its own mean. Besides, the right-hand graph of **??** proposes a representation of the multi-task GPs mixture distribution as a heatmap of probabilities for the location of our predictions. This

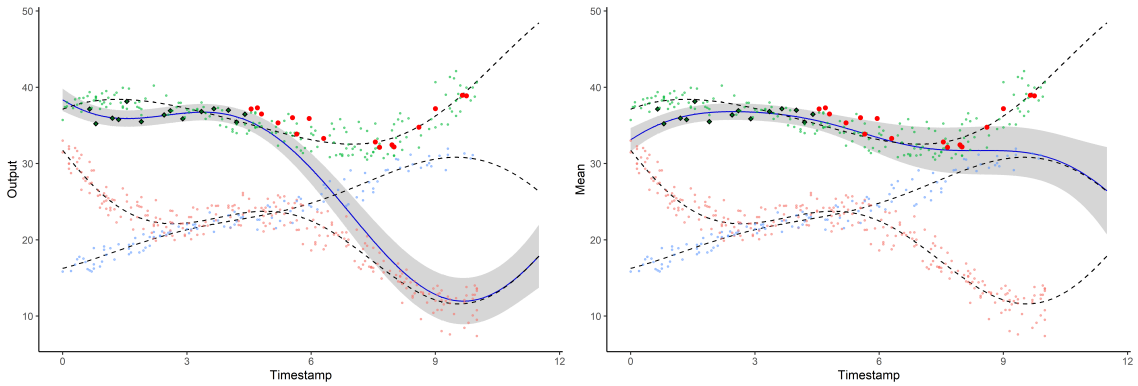

Figure 5: Cluster-specific prediction curves (blue) with associated 95% credible intervals (grey) from MagmaClust, for two unlikely clusters. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training data set, coloured by clusters.

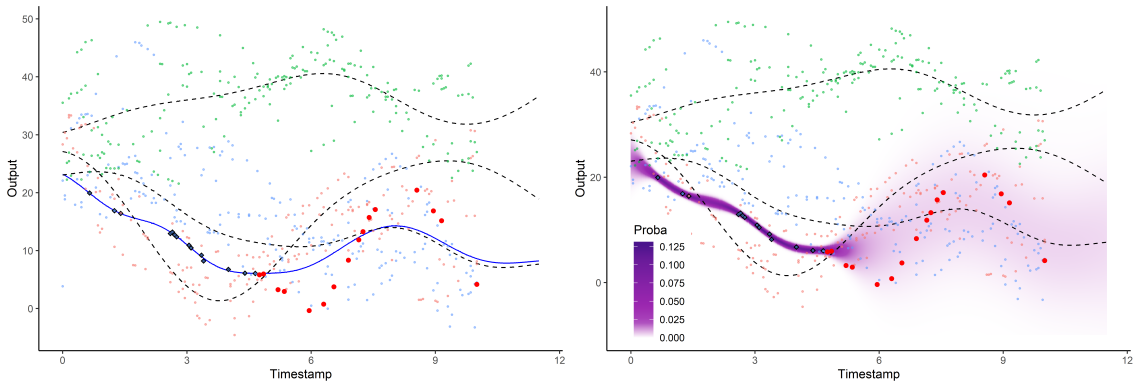

Figure 6: **Left:** GPs mixture mean prediction curve (blue) from MagmaClust. **Right:** heatmap of probabilities for the GPs mixture predictive distribution from MagmaClust. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training data set, coloured by clusters.

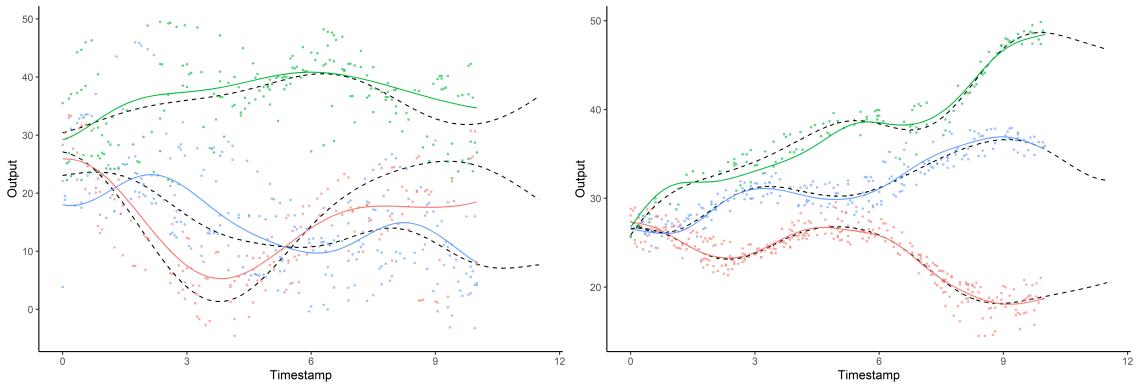

Figure 7: **Left:** fuzzy case. **Right:** well-separated case. Curves of the simulated underlying mean processes, coloured by clusters. The dashed lines represent the mean parameters from the mean processes estimates. Backward points are the observations from the training data set, coloured by clusters.

way, we can display, even in this multi-modal context, a thorough visual quantification for both the dispersion of the predicted values and the confidence we may grant to each of them.

Finally, let us propose on **??** an illustration of the capacity of MagmaClust to retrieve the shape of the underlying mean processes, by plotting their estimations $\{\widehat{m}_k(\cdot)\}_k$ (dotted lines) along with the true curves (plain coloured lines) generated by the simulation scheme. The ability to perform this task generally depends on the structure of the data as well as on the initialisation, although we may observe satisfactory results both on the previous fuzzy example (left) and on a well-separated case (right). It should be noted that the mean processes' estimations also come with uncertainty quantification, albeit not displayed on **??** for the sake of clarity.

6.2 Clustering Performance

Generally, many curve clustering methods struggle to handle irregularly observed data directly. Therefore, for the sake of fairness and to avoid introducing too many smoothing biases in alternative methods, the data sets used in the following are sampled on regular grids, although MagmaClust can deal with irregular measurements by construction (see also **?** for an empirical study). The competing algorithms are the B-splines expansion associated with a kmeans algorithm proposed in **?**, funHDDC (**?**), and funFEM (**?**). The two latter methods were introduced to handle curve clustering problems, by taking advantage of a functional latent mixture modelling, and demonstrated their ability in several applications (**?**). A naive multivariate kmeans is used as initialisation for funHDDC, funFEM, and MagmaClust. We propose on **??** an evaluation of each algorithm in terms of ARI over 100 data sets, simulated from various schemes. First, the procedure detailed in **??** is applied for each of the 4 different hypotheses $(\mathcal{H}_{ki}, \mathcal{H}_{k0}, \mathcal{H}_{0i}, \mathcal{H}_{00})$ to generate data in accordance with our modelling assumptions. Additionally, we propose an alternative simulation scheme, inspired by **?**, to compare performances on data sets that are not tailored for our method. We name this procedure *Scheme A*, and each of the 100 data sets is made of 50 curves, generated randomly, allocated into 4 clusters, and observed at 30 common time points such that:

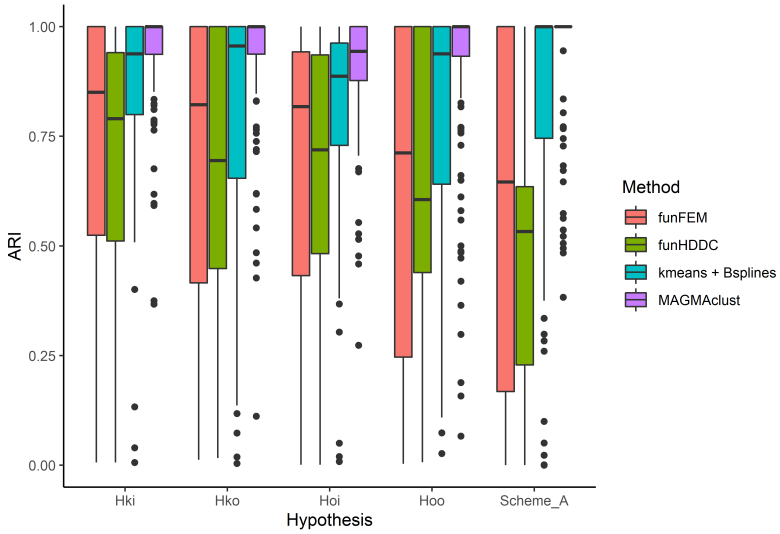

Figure 8: Adjusted rand index (ARI) values between the true clusters and the partitions estimated by kmeans, funHDDC, funFEM, and MagmaClust. The value of K is set to the true number of clusters for all methods. The ARI is computed on 100 data sets for each generating model's assumption $\mathcal{H}_{ki}, \mathcal{H}_{k0}, \mathcal{H}_{0i}$, $\mathcal{H}_{00}$, and *Scheme A*.

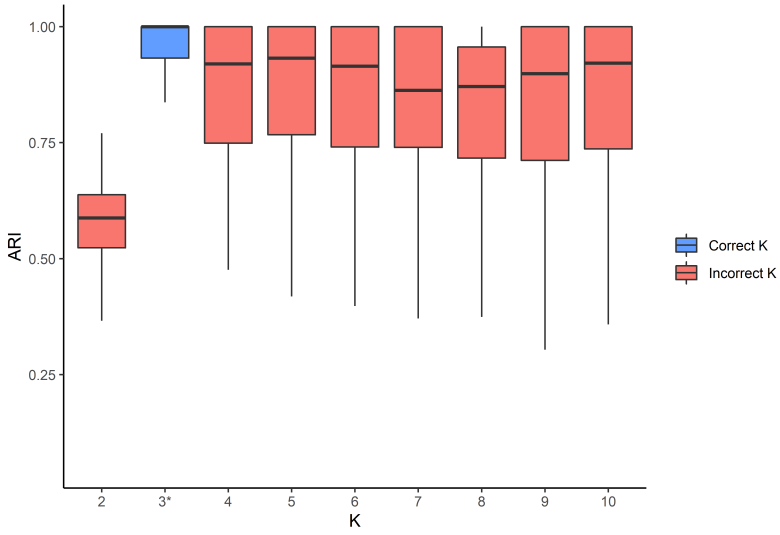

Figure 9: Adjusted Rand Index values between the true clusters and the partitions estimated by MagmaClust with respect to the values of K used as setting. The ARI is computed on the same 100 data sets for each value of K. (3*: the true number of clusters for all data sets)

- Cluster 1: $y(t) = U + 0.5 \times (1 - U) \times (2.5 - |t - 2.5|)^+ + \epsilon, \quad t \in [0, 10]$,
- Cluster 2: $y(t) = U + 0.5 \times (1 - U) \times (2.5 - |t - 7.5|)^+ + \epsilon, \quad t \in [0, 10]$,
- Cluster 3: $y(t) = U + (1 - U) \times (2.5 - |t - 2.5|)^+ + \epsilon, \quad t \in [0, 10]$,
- Cluster 4: $y(t) = U + (1 - U) \times (2.5 - |t - 7.5|)^+ + \epsilon, \quad t \in [0, 10]$,

where $U \sim \mathcal{U}([0, 1])$ and $\epsilon \sim \mathcal{N}(0, 0.05)$.

In all situations, MagmaClust seems to outperform the alternatives . In particular, our approach provides ARI values consistently close to 1 and a lower variance in all contexts. Furthermore, while performances of the other methods are expected to deteriorate because of additional smoothing procedures in the case of irregular grids, MagmaClust would run the same without any change.

On another aspect, **??** provides some insights into the robustness of MagmaClust to an incorrect setting of K, the number of clusters. For 100 data sets with a true value $K^* = 3$, the ARI has been computed between the true partitions and the ones estimated by MagmaClust initialised with different settings $K = 2, \ldots, 10$. Except for $K = 2$ where the low number of clusters prevents from getting enough matching pairs by definition, we may notice relatively unaffected performances as K increases. Despite a non-negligible variance in results, the partitions remain consistent overall, and the clustering performances of MagmaClust seem pretty robust to misspecification of K.

6.3 Model Selection Performance

Table 3 Percentage of data sets for which the true number of cluster is K^*, and the number of cluster selected by the VBIC is K. A total of 50 data sets were simulated for different values $K^* = 1, \ldots, 5$, and MagmaClust was tested on each with varying settings $K = 1, \ldots, 6$, to retain the configuration that reaches the highest VBIC value. The data sets are composed of $M = 50$ (left) or $M = 100$ (right) individuals with $N = 30$ common timestamps, under the hypothesis $\mathcal{H}_{00}$.

	Selected K											
	M = 50						M = 100					
True K*	1	2	3	4	5	6	1	2	3	4	5	6
1	**100**	0	0	0	0	0	**100**	0	0	0	0	0
2	10	**90**	0	0	0	0	2	**96**	2	0	0	0
3	0	22	**74**	2	2	0	2	14	**84**	0	0	0
4	4	10	16	**58**	10	2	2	10	8	**80**	0	0
5	0	10	16	20	**52**	2	0	0	14	18	**68**	0

To remain on the matter of clusters' number, the model selection abilities of the proposed VBIC (**??**) maximisation procedure are assessed on simulated data. For different true numbers of clusters, 50 data sets were simulated according to the previous scheme, and MagmaClust was run successively on each with different settings of K. The setting that reaches the highest value of VBIC is selected as the best model. The percentages of selection for each true K^* and the corresponding values K retained through VBIC are reported in **??**. The procedure seems to adequately select the number of clusters in most context, with results that deteriorate as K grows and a tendency to over-penalise, which

is a classical behaviour with BIC in practice (**?**). As the typical BIC formula relies on asymptotic approximations, we also ran the simulation for different numbers of individuals $M = 50$ and $M = 100$. It may be noticed that the VBIC performs better as M increases, as expected. Such a property appears reassuring since the following real-data applications involves data sets gathering around $M = 10^3$ individuals.

6.4 Prediction Performance

Table 4 Average (sd) values of MSE, $WCIC_{95}$, training and prediction times (in secs) on 100 runs for different numbers of clusters as setting for MagmaClust. (3^* : the true number of clusters for all data sets)

K	MSE	$WCIC_{95}$	Training time	Prediction time
2	7.7 (18.4)	92 (20.3)	70.4 (25)	0.4 (0.1)
3*	3.7 (8.1)	**95 (13.2)**	97.7 (33.2)	0.5 (0.1)
4	3.2 (5.3)	94.9 (13.6)	116.5 (47.3)	0.6 (0.2)
5	3.2 (5.6)	94.4 (14.3)	133 (40.8)	0.6 (0.2)
6	**3.1 (5.4)**	94.4 (13.6)	153.3 (42)	0.8 (0.3)
7	4 (9)	93.6 (15.4)	173.7 (45.1)	1 (0.4)
8	4.7 (13)	93.8 (16)	191.3 (44.7)	1 (0.3)
9	4.1 (9.5)	94 (14.6)	211.6 (52)	0.8 (0.4)
10	4.5 (14.8)	94.4 (14.4)	235 (52.7)	1.8 (1.4)

Another piece of evidence for the robustness to a wrong selection of K is highlighted by **??** in the context of forecasting. The predictive aspect of MagmaClust remains the main purpose of the method and its performances of this task partly rely on the adequate clustering of the individuals. It may be noticed on **??** that both MSE and $WCIC_{95}$ regularly but slowly deteriorate as we move away from the true value of K. However, the performances remain of the same order, and we may still be confident about the predictions obtained through a misspecified running of MagmaClust. In particular, the values of MSE happen to be even better when setting $K = 4, \dots, 6$ (we recall that the same 100 data sets are used in all cases, which can thus be readily compared). Besides, the right-hand part of the table provides indications on the relative time (in seconds) that is needed to train the model for one data set and to make predictions. As expected, both training and prediction times increase roughly linearly with the values of K, which seems consistent with the complexities exposed in **??**. This property is a consequence of the extra mean processes and hyper-parameters that need to be estimated as K grows. Nonetheless, the influence of K is lesser on the prediction time, which yet remains negligible, even when computing many group-specific predictions.

To pursue the matter of prediction, we provide on **??** the comparison of forecasting performances between our method and several state-of-the art alternatives. We use the classical (i.e. single-task) GP regression as benchmark, as we would expect all the competing multi-tasks methods to perform better, considering the additional information that can be shared between individuals. The Magma algorithm (**?**) (which is equivalent to MagmaClust in the specific case where $K = 1$) is also evaluated to measure the accuracy gain we can achieve thanks to the additional clustering on the simulated group-structured data sets. As detailed in **??**, the existing multi-task approaches in the GP literature generally consider specific kernel structures accounting for explicit correlations both between

Table 5 Average (sd) values of MSE, $WCIC_{95}$, training and prediction times (in secs) for GP, SM LMC, MOSM, MAGMA and MAGMACLUST over 100 simulated testing sets.

	MSE	$WCIC_{95}$	Training time	Prediction time
GP	138 (174)	78.4 (31.1)	0 (0)	0.6 (0.1)
SM LMC	29.9 (95.9)	97.2 (7)	1172.6 (300)	5.6 (0.6)
MOSM	417 (646)	57.8 (45.2)	1148 (60)	7.5 (0.5)
MAGMA	31.7 (45)	84.4 (27.9)	61.1 (25.7)	0.5 (0.2)
MAGMACLUST	**3.7 (8.1)**	**95 (13.2)**	132 (55.6)	0.6 (0.2)

inputs and outputs (**?**). The Multi-Output Gaussian Process Tool Kit (MOGPTK) (**?**) is a Python package that implements the main multi-output covariance kernels from literature in a unified framework. We relied on this package to run experiments, first applying the proposed combination (SM LCM) of the spectral mixture (SM) (**?**) and the linear model of coregionalisation (LMC) (**?**), which is the historical and very general formulation for multi-output kernels. Additionally, we made use of the more recent multi-Output spectral mixture (MOSM) algorithm (**?**), which is defined as the default method in the package. For the sake of fair comparison, the L-BFGS-B algorithm (**??**) is the optimisation procedure used in all competing methods.

Regarding both mean prediction and uncertainty quantification, our approach outperforms the alternatives. In terms of MSE, MAGMACLUST takes advantage of its multiple mean processes to results in an order of magnitude enhancement compared to the best competitors (MAGMA and SM LMC). As expected, the simple GP regression performs rather poorly and surprisingly MOSM results are even worse, as in practice, its current implementation seems to reach pathological cases during training most of the time. Additionally, the empirical quantification of uncertainty of MAGMACLUST appears very convincing since there are on average exactly 95% of the observations lying within the weighted CI_{95}, as expected.

In accordance with the theoretical complexity, the increase in training times displayed in **??** remains roughly proportional to the value of K (we recall that MAGMACLUST assumes $K = 3$ here, compared to MAGMA which is $K = 1$). In contrast, the multi-output GPs methods (SM LMC and MOSM) multiply tenfold both training and prediction times even in these reasonable experiments (50 individuals and 30 timestamps). This cost comes from the cubic complexity in the number of tasks (due to size-augmented covariance matrices) that results in a massive computational burden as M increases. As a consequence, in the following real data applications, where thousands of individuals are considered, these methods would merely be unusable in practice.

6.5 Application to real data sets

To evaluate the efficiency of our approach in real-life applications, we introduce 3 distinct data sets corresponding to various contexts: sports performances, weight follow-up during childhood, and missing data reconstruction in CO_2 emissions. The common aspect in all these problems lies in the presence of time series collected from multiple sources, possibly with irregular measurements, for which we expect to provide accurate forecasts by taking advantage of shared information and clustered structures in the data. For all experiments, the individuals (or countries for CO_2) are split into training and testing sets (in proportions

Table 6 Average (sd) values of MSE and $WCIC_{95}$ for MagmaClust with $K = 1, \ldots, 6$ on the french swimmer testing data sets.

	Women		Men	
	Mean	$WCIC_{95}$	Mean	$WCIC_{95}$
Magma	4.9 (6.6)	94.3 (14.2)	3 (3.3)	96.9 (9.3)
MagmaClust- K = 2	4.5 (6.2)	94.3 (14.1)	2.8 (3.2)	96.2 (10.1)
MagmaClust- K = 3	4.3 (6)	**94.4 (14.1)**	2.8 (3.1)	**96.1 (10.4)**
MagmaClust- K = 4	4.2 (5.9)	**94.4 (14.1)**	2.8 (3)	96.2 (10)
MagmaClust- K = 5	4.2 (5.9)	**94.4 (14.1)**	**2.7 (3)**	**96.1 (10.1)**
MagmaClust- K = 6	**4.1 (5.7)**	94.3 (14.1)	2.8 (3)	96.3 (9.8)

60%−40%). In the absence of expert knowledge, the prior mean functions $\{m_k(\cdot)\}_k$ are set to be constant equal to 0. The hypothesis $\mathcal{H}_{00}$ is specified along with random initialisations for the hyper-parameters. The hyper-parameters, the mean processes and the cluster's membership probabilities are learnt on the training data set. Then, the data points of each testing individual are split for evaluation purposes between observed (the first 60%) and testing values (the remaining 40%). Therefore, each new process $y_*(\cdot)$ associated with a test individual is assumed to be partially observed, and its testing values are used to compute MSE and $WCIC_{95}$ for the predictions. As measuring clustering performances directly for real-life applications is a vain effort (**?**), the results are provided for several values of K, from $K = 1$ (i.e. Magma) to $K = 6$, to evaluate the effect of clustered-data structures on predictions. In all the following experiments, predictive performances tend to reach a plateau as we increase the number of clusters and no substantial improvement is noticeable for $K \geq 7$.

Talent identification in competitive swimming

As previously presented in **??**, the 100m freestyle swimming data sets initially proposed in **?** and **?** is analysed below in the new light of MagmaClust. The data sets contain results from 1731 women and 7876 men, members of the French swimming federation, each of them compiling an average of 22.2 data points (min = 15, max = 61) and 12 data points (min = 5, max = 57) respectively. In the following, age of the i-th swimmer is considered as the input variable (timestamp t) and the performance (in seconds) on a 100m freestyle as the output ($y_i(t)$). The analysis focuses on the youth period, from 10 to 20 years, where the progression is the most noticeable. Let us recall that we aim at modelling a curve of progression from competition results for each individual in order to forecast their future performances. We expect MagmaClust to provide relevant predictions by taking advantage of both its multi-task feature and the clustered structure of data, previously highlighted in **?**.

As exhibited by **??**, MagmaClust offers excellent performances on both data sets and slightly improves Magma predictions, as we increase the number of clusters. Values of both MSE and $WCIC_{95}$ appear satisfactory in all cases, and cluster-specific predictions provide additional accuracy though one may fairly argue that the gain remains moderate overall. One of the explaining reasons is highlighted in **??**, where we displayed illustrative predictions for a random man and woman when $K = 5$. Although we can notice clear distinctions between the different mean curves at young ages, these differences tend to decrease afterwards, as adults' performances lie in narrow intervals, especially in regards to

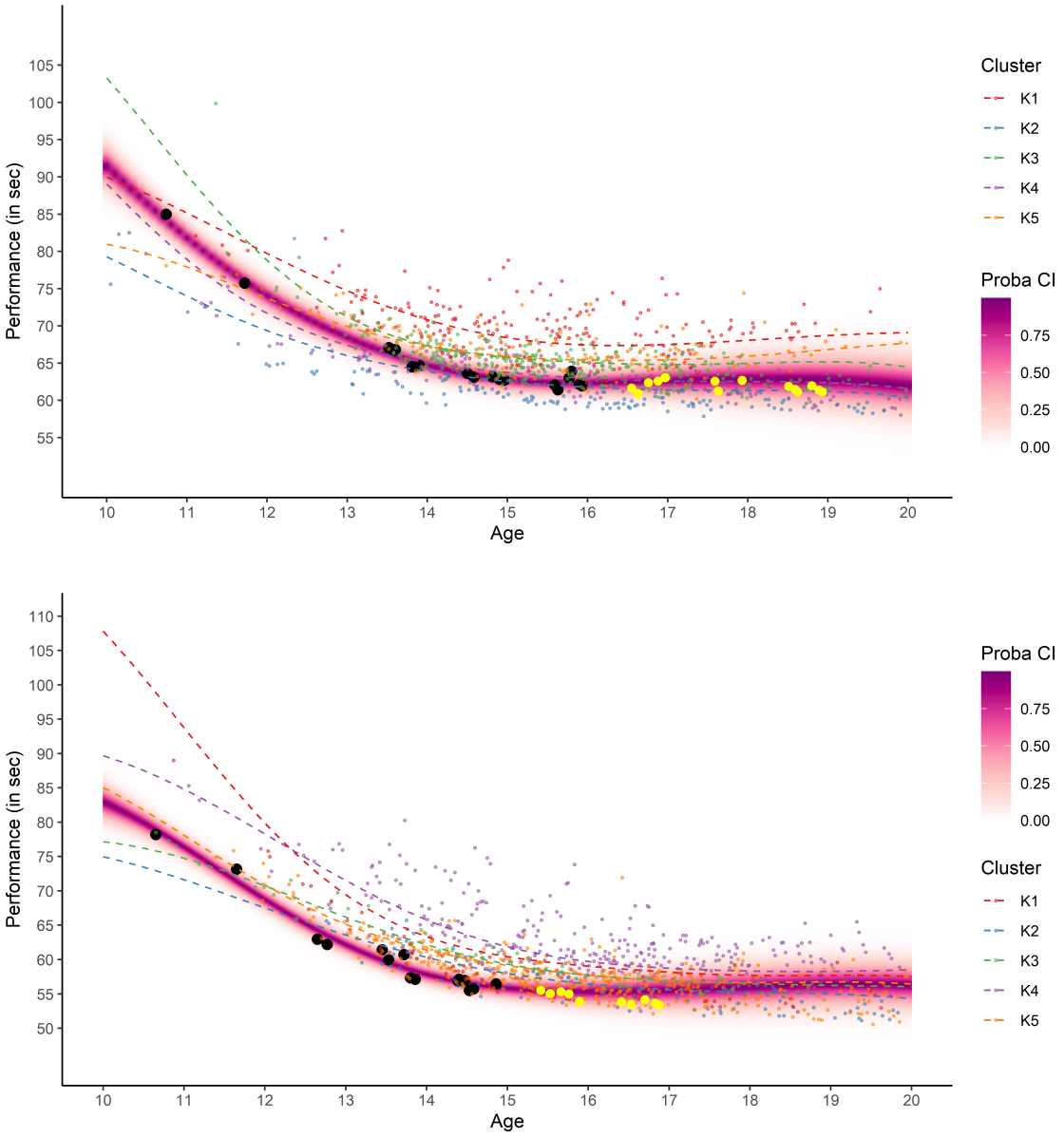

Figure 10: **Top:** women data set. **Bottom:** men data set. Heatmap of probabilities obtained through the GPs mixture predictive distribution of MagmaClust with $K = 5$ for a random illustrative swimmer. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in yellow. Backward points are a sample of observations from the training data set, coloured according to their most probable cluster.

Table 7 Average (sd) values of MSE and $WCIC_{95}$ for MagmaClust with $K = 1, \ldots, 6$ on the children's weight testing data set.

	MSE	$WCIC_{95}$
Magma	5.35 (9.48)	93.3 (16.7)
MagmaClust- K = 2	4.94 (8.98)	**95.1 (15.2)**
MagmaClust- K = 3	5.01 (9.67)	94.7 (16)
MagmaClust- K = 4	4.90 (9)	94.8 (16.1)
MagmaClust- K = 5	4.95 (9.18)	94.7 (16.1)
MagmaClust- K = 6	**4.85 (9.8)**	**94.9 (16.1)**

the overall signal-on-noise ratio. Nevertheless, MagmaClust provides several additional insights into this problem compared to Magma.

First, the clusters offer interesting results to assess the profile of young swimmers and to determine the individuals to whom they most resemble. In particular, it is also possible to differentiate future evolutions associated with each cluster, along with their probabilities to occur (we do not display all the cluster-specific predictions here for the sake of concision). On the other hand, our method leads to tighter predictive distributions in terms of uncertainty. Compared to Magma which uses all training data identically, MagmaClust somehow discards the superfluous information, through the weights τ_{*k}, to only retain the most relevant cluster-specific mean processes.

Follow-up of children's weight in Singapore

In contrast with the previous application, we now study individual time series that are almost similar at young ages before diverging while growing older. This data set (collected through the GUSTO program, see https://www.gusto.sg/) corresponds to a weight follow-up of 342 children from Singapore at 11 timestamps between birth and 72 months. In this experiment, the goal is to predict the weight of a child at $\{24, 36, 48, 60, 72\}$ months, using its observed weight at $\{0, 3, 6, 9, 12, 18\}$ months and data from the training individuals. Since the weight differences between toddlers are shallow until 18 months, providing accurate long-term forecasts, while clear morphology differences emerge, seems particularly challenging at first sight. However, MagmaClust still achieves impressive performances in this context as highlighted by **??**. Once again, the accuracy of predictions seems to slightly improve as we increase the number of clusters. In this application, recovering an adequate cluster for a child is essential to anticipate which future weight pattern is the most likely, and assuming that more clusters exist appears to help in identifying specific patterns more precisely.

This being said, each new cluster increases the overall complexity of the model and whether it is worth adding one cluster regarding the relative gain in accuracy remains a practitioner's choice. For instance, the VBIC measure we proposed in **??** indicates $K = 3$ as the optimal number of clusters for this data set (keep in mind that this criterion maximises a penalised ELBO and tells us nothing regarding the predicting abilities). Using this value for K allows us to display on **??** the behaviour of MagmaClust predictions for a random testing child. Notice that the prediction remains nicely accurate all along even though the testing points (in yellow) are not close to any of the mean processes.

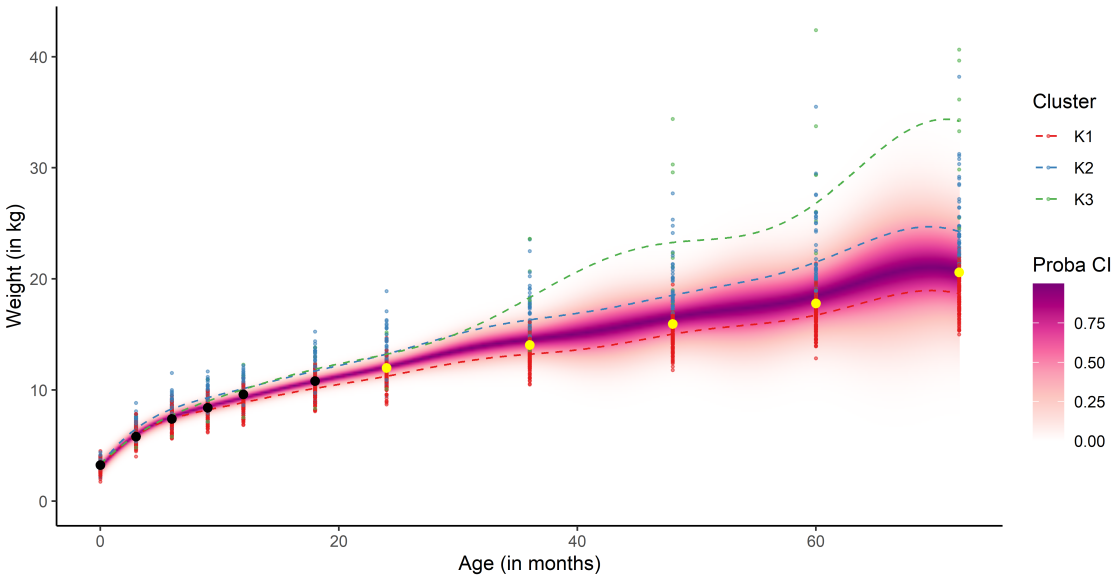

Figure 11: Heatmap of probabilities obtained through the GPs mixture predictive distribution of MagmaClust with $K = 5$ for a random illustrative swimmer. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in yellow. Backward points are observations from the training data set, coloured according to their most probable cluster.

This particularity comes from the learning of cluster weights τ_{*k} at young ages, where the algorithm identifies that this child belonged in nearly equal proportions to the clusters K1 (in red) and K2 (in blue). Therefore, the multi-task GPs mixture posterior distribution defined in **??** defines a weighted trade-off between the two mean processes that remarkably predicts the true weight values of the child for almost 5 years. Although this nice behaviour on a single example does prevent pathological cases to occur in general, we know from **??** that those remain rare in practice, and that the computed credible intervals encompass true data with the correct degree of uncertainty.

Reconstructing missing data in CO_2 emissions

For this last application, we propose to use MagmaClust to tackle a different kind of problem, namely, missing data reconstruction. Contrarily to the previous data sets focused on forecasting, the present collection of time series consists in historical measurements of CO_2 emissions per capita for each country from 1750 to 2020 (freely available at `https://github.com/owid/co2-data`). Naturally, most countries in the world have not collected such data regularly and many annual observations are missing, especially as we move back in the past (for instance, only Canada and the United Kingdom reported values before 1800). However, our method seems particularly well-suited to recover probable distributions of historical CO_2 emissions by exploiting similarities between countries and transferring knowledge from the densely observed time series to those that remain sparse.

To illustrate this ambition, we provide on **??** a visualisation of the raw observations for

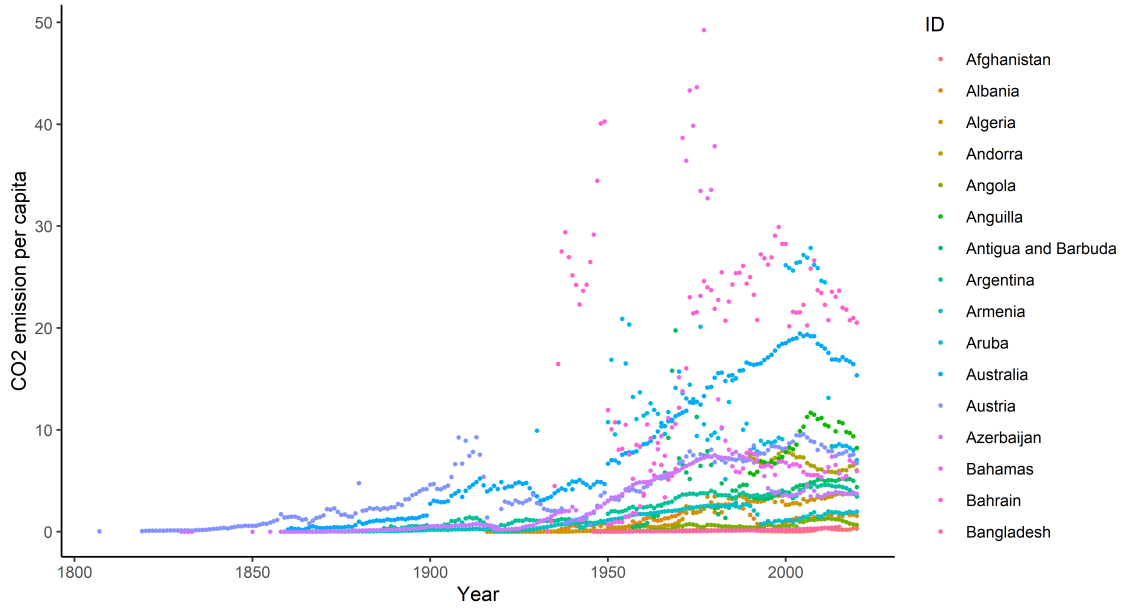

Figure 12: Time series representing the evolution of CO_2 emissions per capita for an illustrative sample of 16 countries, differentiated by colors.

the first 16 countries appearing by alphabetical order in the data set (for the sake of clarity, we cannot display all countries at once). While studying per capita quantities allows us to compare countries with different populations, we can observe that CO_2 emissions can massively differ around the world, and so does the regularity of measurements. By simple inspection, some patterns seem to emerge for countries with similar topographical properties (oil and gas producing regions) or similar lifestyles (typically depending on the average wealth of populations). By applying MagmaClust on this data set, we have noticed its ability to automatically recover patterns that seem *logical* (although such a thing remains highly subjective). For instance, when setting $K = 5$, one cluster of 11 countries gathered United Kingdom, United States, Russia, France and Canada among others. Another cluster only counted Kuwait and Qatar as members, while the largest regrouped around 90 countries mainly from Africa, South Asia and South America, for which the $CO2$ emissions per capita have generally been really low during the past centuries. More generally, we reported the prediction performances on this dataset in **??**. Notice that increasing the number of clusters dramatically improves accuracy in this context, as it seems clear on **??** that some countries present atypical patterns and should be treated in separated clusters to reach satisfying predictions of the missing values. Overall, these abilities to take advantage of group-structured data, even with irregular measurements, and to provide probabilistic statements, highlight once more the interest of MagmaClust to tackle various machine learning problems both in supervised and unsupervised contexts.

7 Discussion

We introduced a novel framework to handle clustering and regression purposes with a multi-task GPs mixture model. This approach, called MagmaClust, extends the previous algorithm Magma (**?**) to deal with group-structured data more efficiently. The

Table 8 Average (sd) values of MSE and $WCIC_{95}$ for MagmaClust with $K = 1, \ldots, 6$ on the CO_2 emissions testing data set.

	Mean	$WCIC_{95}$
Magma	34.9 (89.5)	92.9 (21.6)
MagmaClust- K = 2	28.9 (62.3)	90.8 (24)
MagmaClust- K = 3	19.6 (49.1)	92.5 (18.3)
MagmaClust- K = 4	15.4 (33.4)	93.7 (17.8)
MagmaClust- K = 5	**14 (28.7)**	**94.1 (17.1)**
MagmaClust- K = 6	14.2 (29.3)	93.4 (18)

method provides new insights on the matter of GP regression by introducing cluster-specific modelling and predictions while remaining efficiently tractable through the use of variational approximations for inference. Moreover, this nonparametric probabilistic framework accounts for uncertainty both in regards to the clustering and predictive aspects, which appears to be notable in the machine learning literature. We demonstrated the practical efficiency of MagmaClust on both synthetic and real data sets where it outperformed the alternatives, particularly in group-structured context. Even though the main concern of our method remains the predictive abilities, the clustering performances also deserve to be highlighted, compared to state-of-the-art functional clustering algorithms.

While we recall that computational cost is of paramount importance to ensure broad applicability of GP models, the present version of MagmaClust yet lacks a sparse approximation, which is not trivial to derive in this framework. However, one of the state-of-the-art sparse methods (**??**) also makes use of variational inference, both to select pseudo-inputs and learn hyper-parameters of GP models. Therefore, an interesting extension could come from simultaneously computing $\{\mu_k(\cdot)\}_k$'s hyper-posteriors and pseudo-inputs in MagmaClust, allowing for a sparse approximation of the highest dimensional objects in our model. Besides, several additional features would be worth investigating in future work, such as the extension to non-Gaussian likelihoods or enabling online updates in the learning procedure.

8 Proofs

As a prerequisite, let us introduce an intermediate result that will be used several times in the proofs below.

Lemma 8.1. *Let $X \in \mathbb{R}^N$ be a random Gaussian vector $X \sim \mathcal{N}(m, \mathbf{K})$, where $\mathbb{E}_X$ denotes the expectation and $\mathbb{V}_X$ the variance with respect to this distribution. Additionally, let $b \in \mathbb{R}^N$ be an arbitrary vector and $\mathbf{S}$ a $N \times N$ covariance matrix. Then:*

$$\mathbb{E}_X\left[(X-b)^\intercal \mathbf{S}^{-1}(X-b)\right] = (m-b)^\intercal \mathbf{S}^{-1}(m-b) + tr\left(\mathbf{K}\mathbf{S}^{-1}\right).$$

Proof.

$$\mathbb{E}_X\left[(X-b)^\intercal \mathbf{S}^{-1}(X-b)\right] = \mathbb{E}_X\left[\mathrm{tr}\left(\mathbf{S}^{-1}(X-b)(X-b)^\intercal\right)\right]$$

$$= \operatorname{tr}\left(\mathbf{S}^{-1}(m-b)(m-b)^\intercal\right) + \operatorname{tr}\left(\mathbf{S}^{-1}\mathbb{V}_X\left[X\right]\right)$$
$$= (m-b)^\intercal\mathbf{S}^{-1}(m-b) + \operatorname{tr}\left(\mathbf{K}\mathbf{S}^{-1}\right).$$

□

8.1 Proof of ??

Throughout, we note $\mathbb{E}_{\boldsymbol{\mu}}$ the expectation with respect to the variational distribution $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. From **?**, Chapter 10, the optimal solution $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$ to the variational formulation verifies:

$$\begin{aligned}
\log \widehat{q}_{\mathbf{Z}}(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \widehat{\Theta})\right] + C_1 \\
&= \mathbb{E}_{\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i) + \log p(\mathbf{Z} \mid \widehat{\boldsymbol{\pi}}) + \log p(\boldsymbol{\mu} \mid \{\widehat{\gamma}_k\}_k)\right] + C_1 \\
&= \mathbb{E}_{\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i)\right] + \log p(\mathbf{Z} \mid \widehat{\boldsymbol{\pi}}) + C_2 \\
&= \mathbb{E}_{\boldsymbol{\mu}}\left[\sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik} \log p(\mathbf{y}_i \mid Z_{ik}=1, \mu_k(\mathbf{t}_i), \widehat{\theta}_i, \widehat{\sigma}_i^2)\right] + \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\log(\widehat{\pi}_k) + C_2 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log(\widehat{\pi}_k) + \mathbb{E}_{\mu_k}\left[\log p(\mathbf{y}_i \mid Z_{ik}=1, \mu_k(\mathbf{t}_i), \widehat{\theta}_i, \widehat{\sigma}_i^2)\right]\right] + C_2 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\Bigg[\log(\widehat{\pi}_k) - \frac{1}{2}\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right| \\
&\qquad\qquad - \frac{1}{2}\mathbb{E}_{\mu_k}\left[(\mathbf{y}_i - \mu_k(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}}^{-1}(\mathbf{y}_i - \mu_k(\mathbf{t}_i))\right]\Bigg] + C_3. \qquad (10)
\end{aligned}$$

Applying **??** to the expectation in (**??**), we obtain:

$$\begin{aligned}
\log \widehat{q}_{\mathbf{Z}}(\mathbf{Z}) &= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\Bigg[\log(\widehat{\pi}_k) - \frac{1}{2}\left(\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right| + (\mathbf{y}_i - \widehat{m}_k(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}}^{-1}(\mathbf{y}_i - \widehat{m}_k(\mathbf{t}_i))\right) \\
&\qquad\qquad - \frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}_i}_k {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}}^{-1}\right)\Bigg] + C_3 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log \tau_{ik}\right]
\end{aligned}$$

where (by inspection of both Gaussian and multinomial distributions):

$$\tau_{ik} = \frac{\widehat{\pi}_k \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right)\exp\left(-\frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}_i}_k {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}}^{-1}\right)\right)}{\sum_{l=1}^{K}\widehat{\pi}_l \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_l(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right)\exp\left(-\frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}_i}_l {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}}^{-1}\right)\right)}, \ \forall i \in \mathcal{I}, \forall k \in \mathcal{K}.$$

Therefore, the optimal solution may be written as a factorised form of multinomial distributions:

$$\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{iK})^\intercal\right).$$

8.2 Proof of ??

Let $\mathbb{E}_{\mathbf{Z}}$ denote by the expectation with respect to the variational distribution $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$. From **?**, Chapter 10, the optimal solution $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ to the variational formulation verifies:

$$
\begin{aligned}
\log \widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \mathbb{E}_{\mathbf{Z}}\left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \widehat{\Theta})\right] + C_1 \\
&= \mathbb{E}_{\mathbf{Z}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i) + \log p(\mathbf{Z} \mid \widehat{\boldsymbol{\pi}}) + \log p(\boldsymbol{\mu} \mid \{\widehat{\gamma}_k\}_k)\right] + C_1 \\
&= \mathbb{E}_{\mathbf{Z}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i)\right] + \log p(\boldsymbol{\mu} \mid \{\widehat{\gamma}_k\}_k) + C_2 \\
&= \sum_{i=1}^{M} \mathbb{E}_{\mathbf{Z}_i}\left[\log p(\mathbf{y}_i \mid \mathbf{Z}_i, \boldsymbol{\mu}, \widehat{\theta}_i, \widehat{\sigma}_i^2)\right] + \sum_{k=1}^{K} \log p(\mu_k(\mathbf{t}) \mid \widehat{\gamma}_k) + C_2 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} \mathbb{E}_{\mathbf{Z}_i}\left[Z_{ik}\right] \log p(\mathbf{y}_i \mid Z_{ik} = 1, \mu_k(\mathbf{t}_i), \widehat{\theta}_i, \widehat{\sigma}_i^2) + \sum_{k=1}^{K} \log p(\mu_k(\mathbf{t}) \mid \widehat{\gamma}_k) + C_2 \\
&= -\frac{1}{2}\sum_{k=1}^{K}\Bigg[(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1}(\mu_k(\mathbf{t}) - m_k(\mathbf{t})) \\
&\qquad + \sum_{i=1}^{M} \tau_{ik}(\mathbf{y}_i - \mu_k(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}^{\mathbf{t}_i}}^{-1}(\mathbf{y}_i - \mu_k(\mathbf{t}_i))\Bigg] + C_3.
\end{aligned}
$$

If we regroup the scalar coefficient τ_{ik} with the covariance matrix ${\boldsymbol{\Psi}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}^{\mathbf{t}_i}}^{-1}$, we simply recognise two quadratic terms of Gaussian likelihoods on the variables $\mu_k(\cdot)$, although evaluated onto different sets of timestamps $\mathbf{t}$ and $\mathbf{t}_i$. By taking some writing cautions and expanding the vector-matrix products entirely, it has been proved in **?** that this expression factorises with respect to $\mu_k(\mathbf{t})$ simply by expanding vectors $\mathbf{y}_i$ and matrices $\boldsymbol{\Psi}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}^{\mathbf{t}_i}$ with zeros, $\forall t \in \mathbf{t},\ t \notin \mathbf{t}_i$. Namely, we can notice that:

- $\forall i \in \mathcal{I},\ \tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t)\right)_{t \in \mathbf{t}}$, a N-dimensional vector,
- $\forall i \in \mathcal{I},\ \tilde{\boldsymbol{\Psi}}_i = \left[\mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\widehat{\theta}_i, \widehat{\sigma}_i^2}(t, t')\right]_{t,t' \in \mathbf{t}}$, a $N \times N$ matrix.

Therefore:

$$
\begin{aligned}
\log \widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = -\frac{1}{2}\sum_{k=1}^{K} &\mu_k(\mathbf{t})^\intercal \left({\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} + \sum_{i=1}^{M} \tau_{ik} \tilde{\boldsymbol{\Psi}}_i^{-1}\right) \mu_k(\mathbf{t}) \\
&+ \mu_k(\mathbf{t})^\intercal \left({\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} m_k(\mathbf{t}) + \sum_{i=1}^{M} \tau_{ik} \tilde{\boldsymbol{\Psi}}_i^{-1} \tilde{\mathbf{y}}_i\right) + C_4.
\end{aligned}
$$

By inspection, we recognise a sum of a Gaussian log-likelihoods, which implies the underlying values of the constants. Finally:

$$
\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right), \tag{11}
$$

with:

- $\widehat{\mathbf{C}}_k^{\mathbf{t}} = \left({\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} + \sum_{i=1}^{M} \tau_{ik} \tilde{\mathbf{\Psi}}_i^{-1} \right)^{-1}, \ \forall k \in \mathcal{K},$
- $\widehat{m}_k(\mathbf{t}) = \widehat{\mathbf{C}}_k^{\mathbf{t}} \left({\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} m_k(\mathbf{t}) + \sum_{i=1}^{M} \tau_{ik} \tilde{\mathbf{\Psi}}_i^{-1} \tilde{\mathbf{y}}_i \right), \ \forall k \in \mathcal{K}.$

8.3 Proof of ??

Let $\mathbb{E}_{\mathbf{Z},\boldsymbol{\mu}}$ be the expectation with respect to the optimised variational distributions $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$ and $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. From **?**, Chapter 10, we can figure out the optimal values for the hyper-parameters Θ by maximising the lower bound $\mathcal{L}(\widehat{q}; \Theta)$ with respect to Θ:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{L}(\widehat{q}; \Theta).$$

Moreover, we can develop the formulation of the lower bound by expressing the integrals as an expectation, namely $\mathbb{E}_{\mathbf{Z},\boldsymbol{\mu}}$. Recalling the complete-data likelihood analytical expression and focusing on quantities depending upon Θ, we can write:

$$\begin{aligned}
\mathcal{L}(\widehat{q}; \Theta) &= -\mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}} \left[\underbrace{\log \widehat{q}_{\mathbf{Z},\boldsymbol{\mu}}(\mathbf{Z}, \boldsymbol{\mu})}_{constant\ w.r.t.\ \Theta} - \log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \Theta) \right] \\
&= \mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}} \left[\log \prod_{k=1}^{K} \left\{ \mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right) \prod_{i=1}^{M} \left(\pi_k \mathcal{N}\left(\mathbf{y}_i; \mu_k(\mathbf{t}_i), \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right)\right)^{Z_{ik}} \right\} \right] + C_1 \\
&= \sum_{k=1}^{K} \Bigg[-\frac{1}{2} \Big(\log \left|\mathbf{C}_{\gamma_k}^{\mathbf{t}}\right| + \mathbb{E}_{\boldsymbol{\mu}} \left[(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} (\mu_k(\mathbf{t}) - m_k(\mathbf{t})) \right] \Big) \\
&\qquad - \frac{1}{2} \sum_{i=1}^{M} \mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}} \left[Z_{ik} \left(\log \left| \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i} \right| + (\mathbf{y}_i - \mu_k(\mathbf{t}_i))^\intercal {\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1} (\mathbf{y}_i - \mu_k(\mathbf{t}_i)) \right) \right] \\
&\qquad + \sum_{i=1}^{M} \mathbb{E}_{\mathbf{Z}} \left[Z_{ik} \right] \log \pi_k \Bigg] + C_2 \\
&= \sum_{k=1}^{K} \Bigg[-\frac{1}{2} \Big(\log \left|\mathbf{C}_{\gamma_k}^{\mathbf{t}}\right| + (\widehat{m}_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} (\widehat{m}_k(\mathbf{t}) - m_k(\mathbf{t})) + \operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}} {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} \right) \Big) \\
&\qquad - \frac{1}{2} \sum_{i=1}^{M} \tau_{ik} \left(\log \left| \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i} \right| + (\mathbf{y}_i - \widehat{m}_k(\mathbf{t}_i))^\intercal {\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1} (\mathbf{y}_i - \widehat{m}_k(\mathbf{t}_i)) + \operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}} {\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1} \right) \right) \\
&\qquad + \sum_{i=1}^{M} \tau_{ik} \log \pi_k \Bigg] + C_2,
\end{aligned}$$

where we made use of **??** twice, at the first and second lines for the last equality. By reorganising the terms on the second line, we can derive another formulation of this lower bound that allows for better managing of the computational resources. For information, we give this expression below since it is the quantity implemented in the current version of the MagmaClust code:

$$\mathcal{L}(\widehat{q}; \Theta) = -\frac{1}{2} \sum_{k=1}^{K} \left[\log \left| {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} \right| + (\widehat{m}_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} (\widehat{m}_k(\mathbf{t}) - m_k(\mathbf{t})) + \operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}} {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} \right) \right]$$

$$
\begin{aligned}
&-\frac{1}{2}\sum_{i=1}^{M}\Bigg[\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right| + \mathbf{y}_i^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\mathbf{y}_i - 2\mathbf{y}_i^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\sum_{k=1}^{K}\boldsymbol{\tau}_{ik}\widehat{m}_k(\mathbf{t}_i) \\
&\qquad + \operatorname{tr}\left({\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\sum_{k=1}^{K}\boldsymbol{\tau}_{ik}\left(\widehat{m}_k(\mathbf{t}_i)\widehat{m}_k(\mathbf{t}_i)^\intercal + \widehat{\mathbf{C}}^{\mathbf{t}}_k\right)\right)\Bigg] \\
&+\sum_{k=1}^{K}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik}\left(\log\boldsymbol{\pi}_k\right) + C_2.
\end{aligned}
$$

Regardless of the expression we choose for the following, we can notice that we expressed the lower bound $\mathcal{L}(q;\Theta)$ as a sum where the hyper-parameters $\{\boldsymbol{\gamma}_k\}_k$, $\{\{\boldsymbol{\theta}_i\}_i, \{\sigma_i^2\}_i\}$ and $\boldsymbol{\pi}$ appear in separate terms. Hence, the resulting maximisation procedures are independent of each other. First, we focus on the simplest term that concerns $\boldsymbol{\pi}$, for which we have an analytical update equation. Since there is a constraint on the sum $\sum_{k=1}^{K}\boldsymbol{\pi}_k = 1$, we first need to introduce a Lagrange multiplier in the expression to maximise:

$$
\lambda\left(\sum_{k=1}^{K}\boldsymbol{\pi}_k - 1\right) + \mathcal{L}(q;\Theta). \tag{12}
$$

Setting to 0 the gradient with respect to $\boldsymbol{\pi}_k$ in (**??**), we get:

$$
\lambda + \frac{1}{\boldsymbol{\pi}_k}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik} = 0,\ \forall k \in \mathcal{K}.
$$

Multiplying by $\boldsymbol{\pi}_k$ and summing over k, we deduce the value of λ:

$$
\begin{aligned}
\sum_{k=1}^{K}\boldsymbol{\pi}_k\lambda &= -\sum_{k=1}^{K}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik} \\
\lambda \times 1 &= -\sum_{i=1}^{M} 1 \\
\lambda &= -M.
\end{aligned}
$$

Therefore, the optimal values for $\boldsymbol{\pi}_k$ are expressed as:

$$
\widehat{\boldsymbol{\pi}}_k = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik},\ \forall k \in \mathcal{K}. \tag{13}
$$

Concerning the remaining hyper-parameters, in the absence of analytical optima, we have no choice but to numerically maximise the corresponding terms in $\mathcal{L}(\widehat{q};\Theta)$, namely:

$$
-\frac{1}{2}\sum_{k=1}^{K}\left(\log\left|{\mathbf{C}^{\mathbf{t}}_{\boldsymbol{\gamma}_k}}^{-1}\right| + (\widehat{m}_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}^{\mathbf{t}}_{\boldsymbol{\gamma}_k}}^{-1}(\widehat{m}_k(\mathbf{t}) - m_k(\mathbf{t})) + \operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k {\mathbf{C}^{\mathbf{t}}_{\boldsymbol{\gamma}_k}}^{-1}\right)\right), \tag{14}
$$

and

$$
-\frac{1}{2}\sum_{i=1}^{M}\sum_{k=1}^{K}\boldsymbol{\tau}_{ik}\left(\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right| + (\mathbf{y}_i - \widehat{m}_k(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\mathbf{y}_i - \widehat{m}_k(\mathbf{t}_i)) + \operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right)\right). \tag{15}
$$

It is straightforward to see that some manipulations of linear algebra also allows the derivation of explicit gradients with respect to $\{\gamma_k\}_k$, $\{\theta_i\}_i$ and $\left\{\sigma_i^2\right\}_i$. Hence, we may take advantage of efficient gradient-based methods to handle the optimisation process. We should stress that the quantity (**??**) is a sum on the sole values of k, whereas (**??**) also implies a sum on the values of i. Hence, each term of these sums involves only one hyper-parameter at a time, which thus may be optimised apart from the others. Conversely, if we assume all individuals (respectively all clusters) to share the same set of hyper-parameters, then the full sum has to be maximised upon at once. Therefore, recalling that we introduced 4 different settings according to whether we consider common or specific hyper-parameters for both clusters and individuals, we shall notice the desired maximisation problems that are induced by (**??**) and (**??**).

8.4 Proof of ??

Let us reconsider the expression of $\mathcal{L}(\widehat{q};\Theta)$ from the previous proof. As the model selection procedure takes place after convergence of the learning step, we can use the optimal variational approximation $\widehat{q}_{\mathbf{Z},\boldsymbol{\mu}}$ to compute the lower bound explicitly. Contrarily to the M step though, we now need to develop its full expression, and thus make use of **??** three times.

$$
\begin{aligned}
\mathcal{L}(\widehat{q};\Theta) &= \mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}\left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \Theta) - \log \widehat{q}_{\mathbf{Z},\boldsymbol{\mu}}(\mathbf{Z},\boldsymbol{\mu})\right] \\
&= \mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i)\right] + \mathbb{E}_{\mathbf{Z}}\left[\log p\left(\mathbf{Z} \mid \boldsymbol{\pi}\right)\right] \\
&\quad + \mathbb{E}_{\boldsymbol{\mu}}\left[\log p\left(\boldsymbol{\mu} \mid \{\gamma_k\}_k\right)\right] - \mathbb{E}_{\mathbf{Z}}\left[\log \widehat{q}_{\mathbf{Z}}(\mathbf{Z})\right] - \mathbb{E}_{\boldsymbol{\mu}}\left[\log \widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})\right] \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K}\left\{\tau_{ik}\left(\log \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i)), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}{}^{-1}\right)\right)\right\} \\
&\quad + \sum_{i=1}^{M}\sum_{k=1}^{K}\left\{\tau_{ik}\log\pi_k\right\} + \sum_{k=1}^{K}\left\{\log \mathcal{N}\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\mathbf{C}^{\mathbf{t}}_{\gamma_k}{}^{-1}\right)\right\} \\
&\quad - \sum_{i=1}^{M}\sum_{k=1}^{K}\left\{\tau_{ik}\log\tau_{ik}\right\} - \sum_{k=1}^{K}\left\{\log \mathcal{N}\left(\widehat{m}_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\widehat{\mathbf{C}}_k^{\mathbf{t}\,-1}\right)\right\} \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K}\left\{\tau_{ik}\left(\log \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}{}^{-1}\right)\right)\right\} \\
&\quad + \sum_{k=1}^{K}\left\{\log \mathcal{N}\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\mathbf{C}^{\mathbf{t}}_{\gamma_k}{}^{-1}\right) + \frac{1}{2}\log\left|\widehat{\mathbf{C}}_k^{\mathbf{t}}\right| + N\log 2\pi + N\right\} \\
&\quad + \sum_{i=1}^{M}\sum_{k=1}^{K}\left\{\tau_{ik}\log\frac{\pi_k}{\tau_{ik}}\right\} \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K}\left\{\tau_{ik}\left(\log \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}{}^{-1}\right) + \log\frac{\pi_k}{\tau_{ik}}\right)\right\} \\
&\quad + \sum_{k=1}^{K}\left\{\log \mathcal{N}\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right) - \frac{1}{2}\mathrm{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}}\mathbf{C}^{\mathbf{t}}_{\gamma_k}{}^{-1}\right) + \frac{1}{2}\log\left|\widehat{\mathbf{C}}_k^{\mathbf{t}}\right| + N\log 2\pi + N\right\}.
\end{aligned}
$$

The result follows by considering the analogous expression $\mathcal{L}(\widehat{q};\widehat{\Theta})$ in which the hyper-

parameters are evaluated at their optimal value.

Acknowledgements

Significant parts of this work have been carried out while Arthur Leroy was affiliated with MAP5, Université de Paris, CNRS, UMR 8145, and Department of Computer Science, The University of Sheffield. The authors warmly thank the French Swimming Federation for collecting data and sharing insights on the analysis, as well as Ai Ling Teh and Dennis Wang for providing data from the GUSTO project. The study is supported by the National Research Foundation (NRF) under the Open Fund-Large Collaborative Grant (OF-LCG; MOH-000504) administered by the Singapore Ministry of Health's National Medical Research Council (NMRC) and the Agency for Science, Technology and Research (A*STAR). In RIE2025, GUSTO is supported by funding from the NRF's Human Health and Potential (HHP) Domain, under the Human Potential Programme. Benjamin Guedj acknowledges partial support by the U.S. Army Research Laboratory and the U.S. Army Research Office, and by the U.K. Ministry of Defence and the U.K. Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/R013616/1. Benjamin Guedj acknowledges partial support from the French National Agency for Research, grants ANR-18-CE40-0016-01 and ANR-18-CE23-0015-02.

Data availability

The synthetic data, trained models and results are available at https://github.com/ArthurLeroy/MAGMAclust/tree/master/Simulations. The real data sets, associated trained models and results are available at https://github.com/ArthurLeroy/MAGMAclust/tree/master/Real_Data_Study.

Code availability

The current version of the R package implementing MagmaClust is available on the CRAN and at https://github.com/ArthurLeroy/MagmaClustR.

Bibliography

C. Abraham, P. A. Cornillon, E. Matzner-Løber, and N. Molinari. Unsupervised Curve Clustering using B-Splines. *Scandinavian Journal of Statistics*, 30(3):581–595, September 2003. ISSN 1467-9469. doi: 10.1111/1467-9469.00350.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974. ISSN 1558-2523. doi: 10.1109/TAC.1974.1100705.

Mauricio A. Álvarez and Neil D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes. *Journal of Machine Learning Research*, 12(41):1459–1500, 2011. ISSN 1533-7928.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: A Review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000036.

Hagai Attias. A Variational Baysian Framework for Graphical Models. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding Probabilistic Sparse Gaussian Process Approximations. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1533–1541. Curran Associates, Inc., 2016.

Yoshua Bengio. Gradient-Based Optimization of Hyperparameters. *Neural Computation*, 12(8):1889–1900, August 2000. ISSN 0899-7667. doi: 10.1162/089976600300015187.

J. Bernardo, J. Berger, A. Dawid, and A. Smith. Regression and classification using Gaussian process priors. *Bayesian statistics*, 6:475, 1998.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575, 2003. ISSN 0167-9473. doi: 10.1016/S0167-9473(02)00163-9.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

Edwin V Bonilla, Kian M. Chai, and Christopher Williams. Multi-task Gaussian Process Prediction. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 153–160. Curran Associates, Inc., 2008.

Charles Bouveyron and Julien Jacques. Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300, December 2011. ISSN 1862-5355. doi: 10.1007/s11634-011-0095-6.

Charles Bouveyron, Etienne Côme, and Julien Jacques. The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726–1760, 2015.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. ISBN 978-0-521-83378-3. doi: 10.1017/CBO9780511804441.

Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, July 1997. ISSN 1573-0565. doi: 10.1023/A:1007379606734.

Kai Chen, Twan van Laarhoven, Perry Groot, Jinsong Chen, and Elena Marchiori. Generalized Convolution Spectral Mixture for Multitask Gaussian Processes. *IEEE transactions on neural networks and learning systems*, April 2020. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.2980779.

Vincent Cohen-Addad, Benjamin Guedj, Varun Kanade, and Guy Rom. Online k-means clustering. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics [AISTATS]*, volume 130 of *Proceedings of Machine Learning Research*, pages 1126–1134. PMLR, April 2021. URL http://proceedings.mlr.press/v130/cohen-addad21a.html.

Taco de Wolff, Alejandro Cuevas, and Felipe Tobar. MOGPTK: The multi-output Gaussian process toolkit. *Neurocomputing*, 424:49–53, February 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2020.09.085.

David Duvenaud. *Automatic Model Construction with Gaussian Processes*. Thesis, University of Cambridge, November 2014.

M. Giacofci, S. Lambert-Lacroix, G. Marot, and F. Picard. Wavelet-Based Clustering for Mixed-Effects Functional Models in High Dimension. *Biometrics*, 69(1):31–40, 2013. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2012.01828.x.

Pierre Goovaerts. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.

Kohei Hayashi, Takashi Takenouchi, Ryota Tomioka, and Hisashi Kashima. Self-measuring Similarity for Multi-task Gaussian Process. *Transactions of the Japanese Society for Artificial Intelligence*, 27(3):103–110, 2012. ISSN 1346-8030, 1346-0714. doi: 10.1527/tjsai.27.103.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA, 2013a. AUAI Press.

James Hensman, Neil D Lawrence, and Magnus Rattray. Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC bioinformatics*, 14(1):1–12, 2013b.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, December 1985. ISSN 1432-1343. doi: 10.1007/BF01908075.

Julien Jacques and Cristian Preda. Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171, July 2013. ISSN 09252312. doi: 10.1016/j.neucom.2012.11.042.

Julien Jacques and Cristian Preda. Functional data clustering: A survey. *Advances in Data Analysis and Classification*, 8(3):231–255, September 2014. ISSN 1862-5347, 1862-5355. doi: 10.1007/s11634-013-0158-y.

Huijing Jiang and Nicoleta Serban. Clustering Random Curves Under Spatial Interdependence With Application to Service Accessibility. *Technometrics*, 54(2):108–119, May 2012. ISSN 0040-1706, 1537-2723. doi: 10.1080/00401706.2012.657106.

Jon Kleinberg. An impossibility theorem for clustering. *Advances in neural information processing systems*, 15, 2002.

Arthur Leroy, Andy Marc, Olivier Dupas, Jean Lionel Rey, and Servane Gey. Functional Data Analysis in Sport Science: Example of Swimmers' Progression Curves Clustering. *Applied Sciences*, 8(10):1766, October 2018. doi: 10.3390/app8101766.

Arthur Leroy, Pierre Latouche, Benjamin Guedj, and Servane Gey. Magma: inference and prediction using multi-task gaussian processes with common mean. *Machine Learning*, 111(5):1821–1849, 2022.

Le Li and Benjamin Guedj. Sequential learning of principal curves: Summarizing data streams on the fly. *Entropy*, 23(11), 2021. ISSN 1099-4300. doi: 10.3390/e23111534.

Le Li, Benjamin Guedj, and Sébastien Loustau. A quasi-Bayesian perspective to online clustering. *Electronic Journal of Statistics*, 12(2):3071–3113, 2018. ISSN 1935-7524. doi: 10.1214/18-EJS1479.

José Luis Morales and Jorge Nocedal. Remark on algorithm L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 38(1):7:1–7:4, December 2011. ISSN 0098-3500. doi: 10.1145/2049662.2049669.

Radford M. Neal. Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. *arXiv:physics/9701026*, January 1997.

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. ISSN 0025-5718, 1088-6842. doi: 10.1090/S0025-5718-1980-0572855-7.

Gabriel Parra and Felipe Tobar. Spectral mixture kernels for multi-output gaussian processes. *Advances in Neural Information Processing Systems*, 30, 2017.

J. Quiñonero-Candela, C. E. Rasmussen, and C. K. I. Williams. *Approximation Methods for Gaussian Process Regression*. MIT Press, 2007. ISBN 978-0-262-02625-3.

Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt, and Oliver Stegle. It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1466–1474. Curran Associates, Inc., 2013.

James O. Ramsay and Bernard W. Silverman. *Functional Data Analysis*. Springer, 2005.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9.

Amandine Schmutz, Julien Jacques, Charles Bouveyron, Laurence Cheze, and Pauline Martin. Clustering multivariate functional data in group-specific functional subspaces. *Computational Statistics*, 35(3):1101–1131, 2020.

Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. *Advances in neural information processing systems*, 17, 2004.

Gideon Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, March 1978. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176344136.

J. Q. Shi and B. Wang. Curve prediction and clustering with mixtures of Gaussian process functional regression models. *Statistics and Computing*, 18(3):267–283, 2008. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-008-9055-1.

J. Q. Shi, B. Wang, R. Murray-Smith, and D. M. Titterington. Gaussian Process Functional Regression Modeling for Batch Data. *Biometrics*, 63(3):714–723, 2007. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2007.00758.x.

J.Q. Shi, R. Murray-Smith, and D.M. Titterington. Hierarchical Gaussian process mixtures for regression. *Statistics and Computing*, 15(1):31–41, 2005. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-005-4787-7.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18, 2005.

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-Task Bayesian Optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2004–2012. Curran Associates, Inc., 2013.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

Naonori Ueda and Ryohei Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282, March 1998. ISSN 0893-6080. doi: 10.1016/S0893-6080(97)00133-0.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. GPstuff: Bayesian Modeling with Gaussian Processes. *Journal of Machine Learning Research*, 14(Apr):1175–1179, 2013. ISSN ISSN 1533-7928.

David L. Weakliem. A Critique of the Bayesian Information Criterion for Model Selection:. *Sociological Methods & Research*, June 2016. doi: 10.1177/0049124199027003002.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of Machine Learning Research*, volume 28, pages 1067–1075, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Jingjing Yang, Hongxiao Zhu, Taeryon Choi, and Dennis D. Cox. Smoothing and Mean–Covariance Estimation of Functional Data with a Bayesian Hierarchical Model. *Bayesian Analysis*, 11(3):649–670, 2016. ISSN 1936-0975, 1931-6690. doi: 10.1214/15-BA967.

Chong You, John T. Ormerod, and Samuel Müller. On Variational Bayes Estimation and Variational Information Criteria for Linear Regression Models. *Australian & New Zealand Journal of Statistics*, 56(1):73–87, 2014. ISSN 1467-842X. doi: 10.1111/anzs.12063.

Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning Gaussian Processes from Multiple Tasks. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 1012–1019, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-180-1. doi: 10.1145/1102351.1102479.