Article:

Leroy, A., Latouche, P., Guedj, B. et al. (1 more author) (2022) MAGMA : inference and prediction using multi-task Gaussian processes with common mean. Machine Learning. ISSN 0885-6125

https://doi.org/10.1007/s10994-022-06172-1

Machine Learning
https://doi.org/10.1007/s10994-022-06172-1

MAGMA: inference and prediction using multi-task Gaussian processes with common mean

Arthur Leroy[1] · Pierre Latouche[2] · Benjamin Guedj[3,4] · Servane Gey[2]

Received: 12 April 2021 / Revised: 18 January 2022 / Accepted: 30 March 2022

Abstract
A novel multi-task Gaussian process (GP) framework is proposed, by using a common mean process for sharing information across tasks. In particular, we investigate the problem of time series forecasting, with the objective to improve multiple-step-ahead predictions. The common mean process is defined as a GP for which the hyper-posterior distribution is tractable. Therefore an EM algorithm is derived for handling both hyper-parameters optimisation and hyper-posterior computation. Unlike previous approaches in the literature, the model fully accounts for uncertainty and can handle irregular grids of observations while maintaining explicit formulations, by modelling the mean process in a unified GP framework. Predictive analytical equations are provided, integrating information shared across tasks through a relevant prior mean. This approach greatly improves the predictive performances, even far from observations, and may reduce significantly the computational complexity compared to traditional multi-task GP models. Our overall algorithm is called Magma (standing for Multi tAsk GPs with common MeAn). The quality of the mean process estimation, predictive performances, and comparisons to alternatives are assessed in various simulated scenarios and on real datasets.

Keywords Multi-task learning · Gaussian processes · EM algorithm · Common mean process · Functional data analysis

1 Introduction

Gaussian processes (GPs) are a powerful tool, widely used in machine learning (Bishop, 2006; Rasmussen & Williams, 2006). The classic context of regression aims at inferring the underlying mapping function associating input to output data. In a probabilistic framework, a typical strategy is to assume that this function is drawn from a prior GP. Doing so, we may enforce some properties for the function solely by characterising the mean and

Editor: Ulf Brefeld.

✉ Arthur Leroy
arthur.leroy.pro@gmail.com
https://arthur-leroy.netlify.app/

Extended author information available on the last page of the article

Published online: 06 May 2022
Springer

covariance functions of the process, the latter often being associated with a specific kernel. This covariance function plays a central role and GPs are an example of kernel methods. We refer to Álvarez et al. (2012) for a comprehensive review. On the other hand, the mean function is generally set to 0 for all entries assuming that the covariance structure already integrates the desired relationship between observed data and prediction targets. In this paper, we consider a novel multi-task learning framework where a series of GPs share a common mean, expressed as a GP as well. We demonstrate that modelling the mean function as such can be key to obtain more relevant predictions.

Related work

The multi-task framework consists in using data from several tasks (or individuals) to improve learning or predictive capacities compared to an isolated model. It has been introduced by Caruana (1997) and then adapted in many fields of machine learning. GP versions of such models were introduced in Schwaighofer et al. (2004), which proposed an Expectation-Maximisation (EM) algorithm for learning. Similar techniques can be found in Shi et al. (2005). Meanwhile, Yu et al. (2005) offered an extensive study of the relationships between the linear model and GPs to develop a multi-task GP formulation. However, since the introduction in Bonilla et al. (2008) of the idea of two matrices, modelling covariance between inputs and tasks respectively, the term *multi-task Gaussian process* has mostly referred to the choice made regarding the covariance structure. Some further developments were discussed by Hayashi et al. (2012), Rakitsch et al. (2013) and Zhu & Sun (2014). In particular, an interesting approach in Nguyen and Bonilla (2014) proposed a sparse approximation for multi-task GP inference. More generally, these approaches are known as examples of *linear models of coregionalization* (LMC) in the geostatistics literature, and Álvarez & Lawrence (2011) provides a unified view on the topic as well as an efficient strategy for constructing computationally efficient approximations. Let us emphasise that the present paper is not based on the same assumptions and principles, and aims at defining a different multi-task paradigm for GPs, focusing on sharing information through the mean function rather than the covariance structure. Besides, the work of Swersky et al. (2013) on Bayesian hyper-parameter optimisation in such LMC models is also worth a mention. Real applications were tackled by similar models in Williams et al. (2009) and Alaa & van der Schaar (2017), while Clingerman & Eaton (2017) and Moreno-Muñoz et al. (2019) developed continual learning methods for multi-task GP.

As we focus on multi-task time series forecasting, a connection can be drawn to the study of multiple curves, or functional data analysis (FDA). As initially proposed in Rice & Silverman (1991), it is possible to model and learn mean and covariance structures simultaneously in this context. We refer to the monographs Ramsay & Silverman (2005) and Ferraty & Vieu (2006) for a comprehensive introduction to FDA. In particular, these books introduced several usual ways for modelling a set of functional objects in frequentist frameworks, for example by using a decomposition in a basis of functions (such as B-splines, wavelets, Fourier). This kind of B-splines decomposition was used in Shi et al. (2007) for modelling the mean function in a generative model that somehow resembles ours. Subsequently, some Bayesian alternatives were developed in Thompson & Rosen (2008), and Crainiceanu & Goldsmith (2010).

Our contributions

A multi-task GP framework with a common mean process is introduced, allowing reliable probabilistic forecasts even in multiple-step-ahead problems, or for sparsely observed individuals. For this purpose, (i) we introduce a GP model where the specific covariance structure of each task is defined through a separate kernel and its associated set of hyper-parameters, whereas the common mean function μ_0 allows sharing information across tasks and overcomes the weaknesses of classic GPs in making predictions far from observed data. To account for uncertainty, we propose a hierarchical formulation to define the common mean process μ_0 as a GP as well. (ii) We derive an algorithm called Magma (available as an R package at https://github.com/ArthurLeroy/MagmaClustR) to compute μ_0's hyper-posterior distribution together with the estimation of hyper-parameters in an EM fashion, and discuss its computational complexity. (iii) We enrich Magma with explicit formulas to make predictions for any new, partially observed, task. The hyper-posterior distribution of μ_0 provides a prior belief on what we would expect to observe before seeing any new data, acting as an already-informed mean process, integrating both trend and uncertainty coming from other tasks. (iv) We illustrate the performance of our method on synthetic and two real-life datasets and obtain state-of-the-art results compared to alternative approaches.

Outline

The paper is organised as follows. We introduce our multi-task Gaussian process model in Sect. 2, along with notation. Section 3 is devoted to the inference procedure, with an Expectation-Maximisation (EM) algorithm to estimate the Gaussian process hyper-parameters and μ_0's hyper-posterior. We leverage this strategy in Sect. 4 and derive a prediction algorithm. In Sect. 5, we analyse and discuss the computational complexity of both the inference and prediction procedures. Our methodology is illustrated in Sect. 6, with a series of experiments on both synthetic and real-life datasets, and a comparison to competing state-of-the-art algorithms. On those tasks, we provide empirical evidence that our algorithm outperforms other approaches. Section 7 draws perspectives for future work, and we defer some proofs to original results claimed in the paper to Sect. 8.

2 The model

2.1 Notation

While GPs can handle many types of data, their continuous nature makes them particularly well suited to study temporal phenomena. Throughout, the term *individual* is used as a synonym of *task* or *batch*, and we adopt notation and vocabulary of time series to remain consistent with the application on real dataset provided in Sect. 6.5, which addresses young swimmers performances' forecast.

We are provided with functional data coming from $M \in \mathcal{I}$ different individuals, where $\mathcal{I} \subset \mathbb{N}$. For each individual i, we observe a set of inputs $\{t_i^1, \dots, t_i^{N_i}\}$ and associated outputs $\{y_i(t_i^1), \dots, y_i(t_i^{N_i})\}$, where N_i is the number of data points for the i-th individual. Since many objects are defined for all individuals, we shorten our notation as follows: for any object x existing for all i, we denote $\left\{x_i\right\}_i = \left\{x_1, \dots, x_M\right\}$. Moreover, as we work in a temporal context, the inputs are referred to as *timestamps*. In the specific case where all individuals are observed at the same timestamps, we call the grid of observations *common*. On the contrary, a grid of observations is *uncommon* if the timestamps are different in number and/or location among the individuals. Some convenient notation follows:

- $\mathbf{t}_i = \{t_i^1, \dots, t_i^{N_i}\}$, 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,
- $\mathbf{t} = \bigcup\limits_{i=1}^{M} \mathbf{t}_i$, the pooled set of timestamps among individuals,
- $N = \text{card}(\mathbf{t})$, the total number of observed timestamps.

2.2 Model and hypotheses

Suppose that functional data are coming from the sum of a mean process, common to all individuals, and an individual-specific centred process. To clarify relationships in the generative model, we illustrate our graphical model in Fig. 1. Let $\mathcal{T}$ be the input space, our model is

$$y_i(t) = \mu_0(t) + f_i(t) + \epsilon_i(t), \ \ \forall t \in \mathcal{T}, \ \ \forall i \in \mathcal{I},$$

where $\mu_0(\cdot) \sim \mathcal{GP}(m_0(\cdot), k_{\theta_0}(\cdot,\cdot))$ and $f_i(\cdot) \sim \mathcal{GP}\big(0, c_{\theta_i}(\cdot,\cdot)\big)$ are respectively the common mean and individual specific processes. Moreover, the error term is supposed to be $\epsilon_i(\cdot) \sim \mathcal{N}(0, \sigma_i^2 I)$. The following notation is used for parameters:

- $m_0(\cdot)$, an arbitrary prior mean function,
- $k_{\theta_0}(\cdot,\cdot)$, a covariance kernel of hyper-parameters θ_0,
- $\forall i \in \mathcal{I}, \ c_{\theta_i}(\cdot,\cdot)$, a covariance kernel with hyper-parameters θ_i,
- $\sigma_i^2 \in \mathbb{R}^+$, the noise variance associated with the i-th individual,
- $\forall i \in \mathcal{I}$, we define the shorthand $\psi_{\theta_i,\sigma_i^2}(\cdot,\cdot) = c_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I$,
- $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$, the set of all hyper-parameters to learn in the model.

We also assume that:

- $\{f_i\}_i$ are independent,
- $\{\epsilon_i\}_i$ are independent,
- $\forall i \in \mathcal{I}, \ \mu_0, f_i$ and ϵ_i are independent.

It follows that $\{y_i \mid \mu_0\}_{i=1,\dots,M}$ are independent from one another, and for all $i \in \mathcal{I}$:

$$y_i(\cdot) \mid \mu_0(\cdot) \sim \mathcal{GP}(\mu_0(\cdot), \psi_{\theta_i,\sigma_i^2}(\cdot,\cdot)).$$

Let us emphasise that this property only holds conditionally to μ_0. Otherwise, once μ_0 is integrated out, the y_i are no longer independent. Here, we do not assume any specific covariance structure between individuals contrarily to standard LMC approaches. As we shall see in the next sections, the process μ_0 will be key to handle the dependencies and share information across the individuals.

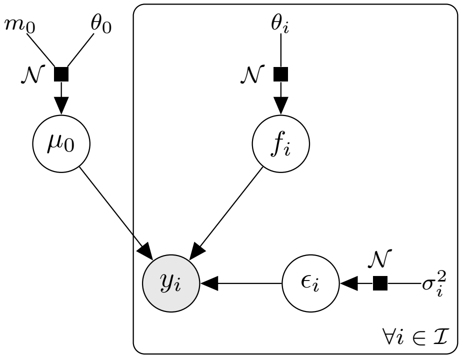

Fig. 1 Graphical model of dependencies between variables in the Multi-task Gaussian Process model

Although this model is based on infinite-dimensional GPs, the inference will be conducted on a finite grid of observations. According to the aforementioned notation, we observe $\{(\mathbf{t}_i, \mathbf{y}_i)\}_i$, and the corresponding likelihoods are Gaussian:

$$\mathbf{y}_i \mid \mu_0(\mathbf{t}_i) \sim \mathcal{N}(\mathbf{y}_i; \mu_0(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}),$$

where $\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2} = \psi_{\theta_i,\sigma_i^2}(\mathbf{t}_i, \mathbf{t}_i) = \left[\psi_{\theta_i,\sigma_i^2}(k, l)\right]_{k,\ell \in \mathbf{t}_i}$ is a $N_i \times N_i$ covariance matrix. Since $\mathbf{t}_i$ might be different among individuals, we also need to evaluate μ_0 on the pooled grid of timestamps $\mathbf{t}$:

$$\mu_0(\mathbf{t}) \sim \mathcal{N}\Big(\mu_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\Big),$$

where $\mathbf{K}^{\mathbf{t}}_{\theta_0} = k_{\theta_0}(\mathbf{t}, \mathbf{t}) = \left[k_{\theta_0}(k, \ell)\right]_{k,l \in \mathbf{t}}$ is a $N \times N$ covariance matrix.

An alternative hypothesis consists in considering hyper-parameters $\{\theta_i\}_i$ and $\{\sigma_i^2\}_i$ equal for all individuals. We call this hypothesis *Common HP* (where *HP* stands for *hyper-parameters*) in the Sect. 6. This particular case represents a context where individuals correspond to different trajectories of the same process, whereas different hyper-parameters indicate different covariance structures and thus a more flexible model. For the sake of generality, the remainder of the paper is written with θ_i and σ_i^2 notation, when there are no differences in the procedure. Moreover, the model above and the subsequent algorithm may use any form of covariance function, often parametrised by a finite set (usually small) of hyper-parameters. For example, a common kernel in the GP literature is known as the *Exponentiated Quadratic* kernel (also called sometimes Squared Exponential or Radial Basis Function kernel). It solely depends on two hyper-parameters $\theta = \{v, \ell\}$ and is defined as:

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

The *Exponentiated Quadratic* kernel is simple and enjoys useful smoothness properties. This is the kernel used in the current version of our implementation (see Sect. 6 for details). Note that there is a rich literature on kernel choice, their construction and properties, which is beyond the scope of the present work: we refer to Rasmussen and Williams (2006) or Duvenaud (2014) for comprehensive studies.

3 Inference

3.1 Learning

Several approaches to learn hyper-parameters for Gaussian processes have been proposed in the literature, we refer to Rasmussen and Williams (2006) for a comprehensive study. One classical approach, called *empirical Bayes* (Casella 1985), is based on the maximisation of an explicit likelihood to estimate hyper-parameters. This procedure avoids sampling from intractable distributions, usually resulting in additional computational cost and complicating practical use in moderate to large sample sizes. As previously stated, once μ_0 is marginalised out, the log-likelihood cannot be written as a sum of Gaussian log-likelihoods any more. Therefore, we propose an EM algorithm (see the pseudocode in Algorithm 1) to learn the hyper-parameters Θ in this context. The procedure alternatively computes the hyper-posterior distribution $p(\mu_0 \mid (\mathbf{y}_i)_i, \widehat{\Theta})$ with current hyper-parameters, and then optimises Θ according to this hyper-posterior distribution. This EM algorithm converges to local maxima (Dempster et al. 1977), typically in a handful of iterations.

E step

For the sake of simplicity, we assume in that section that $\forall i, j \in \mathcal{I}$, $\mathbf{t}_i = \mathbf{t}_j = \mathbf{t}$, i.e. the individuals are observed on a common grid of timestamps. We provide a generalisation of the following proposition in Sect. 4 (Proposition 4), where the result holds for uncommon grids. The E step then consists in computing the hyper-posterior distribution of $\mu_0(\mathbf{t})$.

Proposition 1 *Assume the hyper-parameters $\widehat{\Theta}$ known from initialisation or estimated from a previous M step. The hyper-posterior distribution of μ_0 remains Gaussian*:

$$p\Big(\mu_0(\mathbf{t}) \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}\Big) = \mathcal{N}\Big(\mu_0(\mathbf{t}); \widehat{m}_0(\mathbf{t}), \widehat{\mathbf{K}}^{\mathbf{t}}\Big), \tag{2}$$

with

- $\widehat{\mathbf{K}}^{\mathbf{t}} = \left({\mathbf{K}^{\mathbf{t}}_{\widehat{\theta}_0}}^{-1} + \sum\limits_{i=1}^{M} {\boldsymbol{\Psi}^{\mathbf{t}}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}}^{-1} \right)^{-1}$,
- $\widehat{m}_0(\mathbf{t}) = \widehat{\mathbf{K}}^{\mathbf{t}} \left({\mathbf{K}^{\mathbf{t}}_{\widehat{\theta}_0}}^{-1} m_0(\mathbf{t}) + \sum\limits_{i=1}^{M} {\boldsymbol{\Psi}^{\mathbf{t}}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}}^{-1} \mathbf{y}_i \right)$.

Proof We omit specifying timestamps in what follows since each process is evaluated on $\mathbf{t}$. Therefore, we can write:

$$\begin{aligned} p\Big(\mu_0 \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}\Big) &\propto p\Big(\{\mathbf{y}_i\}_i \mid \mu_0, \widehat{\Theta}\Big) p\Big(\mu_0 \mid \widehat{\Theta}\Big) \\ &\propto \left\{ \prod_{i=1}^{M} p\Big(\mathbf{y}_i \mid \mu_0, \widehat{\theta}_i, \widehat{\sigma}_i^2\Big) \right\} p\Big(\mu_0 \mid \widehat{\theta}_0\Big) \\ &\propto \left\{ \prod_{i=1}^{M} \mathcal{N}\Big(\mathbf{y}_i; \mu_0, \boldsymbol{\Psi}_{\widehat{\theta}_i, \widehat{\sigma}_i^2})\Big) \right\} \mathcal{N}\Big(\mu_0; m_0, \mathbf{K}_{\widehat{\theta}_0}\Big). \end{aligned}$$

The term $\mathcal{L}_1 = -(1/2) \log p(\mu_0 \mid \{\mathbf{y}_i\}_i, \widehat{\Theta})$ may then be written as

$$\begin{aligned} \mathcal{L}_1 &= -\frac{1}{2} \log p(\mu_0 \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}) \\ &= \sum_{i=1}^{M} \big(y_i - \mu_0\big)^{\intercal} \boldsymbol{\Psi}^{-1}_{\widehat{\theta}_i, \widehat{\sigma}_i^2} \big(y_i - \mu_0\big) + \big(\mu_0 - m_0\big)^{\intercal} \mathbf{K}^{-1}_{\widehat{\theta}_0} \big(\mu_0 - m_0\big) + C_1 \\ &= \sum_{i=1}^{M} \mu_0^{\intercal} \boldsymbol{\Psi}^{-1}_{\widehat{\theta}_i, \widehat{\sigma}_i^2} \mu_0 - 2\mu_0^{\intercal} \boldsymbol{\Psi}^{-1}_{\widehat{\theta}_i, \widehat{\sigma}_i^2} \mathbf{y}_i + \mu_0^{\intercal} \mathbf{K}^{-1}_{\widehat{\theta}_0} \mu_0 - 2\mu_0^{\intercal} \mathbf{K}^{-1}_{\widehat{\theta}_0} m_0 + C_2 \\ &= \mu_0^{\intercal} \left(\mathbf{K}^{-1}_{\widehat{\theta}_0} + \sum_{i=1}^{M} \boldsymbol{\Psi}^{-1}_{\widehat{\theta}_i, \widehat{\sigma}_i^2} \right) \mu_0 - 2\mu_0^{\intercal} \left(\mathbf{K}^{-1}_{\widehat{\theta}_0} m_0 + \sum_{i=1}^{M} \boldsymbol{\Psi}^{-1}_{\widehat{\theta}_i, \widehat{\sigma}_i^2} \mathbf{y}_i \right) + C_2, \end{aligned}$$

where the constant terms are gathered into $C_1, C_2 \in \mathbb{R}$. Identifying terms in the quadratic form with the Gaussian likelihood, we get the desired result. □

The maximisation step depends on the assumptions on the generative model, resulting in two versions for the EM algorithm (the E step is common to both, the branching point is here).

M step: different hyper-parameters

Assuming each individual has its own set of hyper-parameters $\{\theta_i, \sigma_i^2\}$, the M step is given by the following procedure.

Proposition 2 *Assume* $p(\mu_0 \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\Big(\mu_0(\mathbf{t}); \widehat{m}_0(\mathbf{t}), \widehat{\mathbf{K}}^{\mathbf{t}}\Big)$ *computed in a previous E step. For a set of hyper-parameters* $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$, *optimal values are given by*

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta)\right],$$

inducing $M + 1$ *independent maximisation problems*:

$$\widehat{\theta}_0 = \underset{\theta_0}{\operatorname{argmax}}\ \mathcal{L}^{\mathbf{t}}\left(\widehat{m}_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right),$$

$$(\widehat{\theta}_i, \widehat{\sigma}_i^2) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}}\ \mathcal{L}^{\mathbf{t}_i}(\mathbf{y}_i; \widehat{m}_0(\mathbf{t}), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}),\ \forall i,$$

where

$$\mathcal{L}^{\mathbf{t}}(\mathbf{x}; \mathbf{m}, \mathbf{S}) = \log \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{S}) - \frac{1}{2}\operatorname{Tr}\left(\widehat{\mathbf{K}}^{\mathbf{t}} \mathbf{S}^{-1}\right).$$

Proof One simply has to distribute the conditional expectation in order to get the right likelihood to maximise, and then notice that the function can be written as a sum of $M + 1$ independent (with respect to the hyper-parameters) terms. Moreover, by rearranging, one can observe that each independent term is the sum of a Gaussian likelihood and a correction trace term. See Sect. 8.2 for details. □

M step: common hyper-parameters

Alternatively, assuming all individuals share the same set of hyper-parameters $\{\theta, \sigma^2\}$, the M step is given by the following procedure.

Proposition 3 *Assume* $p(\mu_0 \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\Big(\mu_0(\mathbf{t}); \widehat{m}_0(\mathbf{t}), \widehat{\mathbf{K}}^{\mathbf{t}}\Big)$*computed in a previous E step. For a set of hyper-parameters* $\Theta = \{\theta_0, \theta, \sigma^2\}$, *optimal values are given by*

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta)\right],$$

inducing two independent maximisation problems:

$$\widehat{\theta}_0 = \underset{\theta_0}{\operatorname{argmax}}\ \mathcal{L}^{\mathbf{t}}\left(\widehat{m}_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right),$$

$$(\widehat{\theta}, \widehat{\sigma}^2) = \underset{\theta, \sigma^2}{\operatorname{argmax}}\ \mathcal{L}_M(\theta, \sigma^2),$$

where

$$\mathcal{L}_M(\theta, \sigma^2) = \sum_{i=1}^{M} \mathcal{L}^{\mathbf{t}_i}(\mathbf{y}_i; \widehat{m}_0(\mathbf{t}), \boldsymbol{\Psi}_{\theta,\sigma^2}^{\mathbf{t}_i}).$$

Proof We use the same strategy as for Proposition 2, see Sect. 8.2 for details. □

In both cases, explicit gradients associated with the likelihoods to maximise are available, facilitating the optimisation with gradient-based methods.

3.2 Initialisation

To implement the EM algorithm described above, several constants must be (appropriately) initialised:

- $m_0(\cdot)$, the mean parameter from the hyper-prior distribution of the process $\mu_0(\cdot)$. A somewhat classical choice in GP is to set its value to a constant function, typically 0 in the absence of external knowledge. Notice that, in our multi-task framework, the influence of $m_0(\cdot)$ in hyper-posterior computation decreases as M grows anyway (see Proposition 1).
- Initial values for kernel parameters θ_0 and $\left\{\theta_i\right\}_i$. Those strongly depend on the chosen kernel and its properties. We advise initiating θ_0 and $\left\{\theta_i\right\}_i$ with close values, as a too large difference might induce nearly singular covariance matrices and result in numerical instability (typical in GPs applications). In such pathological regime, the influence of a specific individual tends to overtake others in the calculus of μ_0's hyper-posterior distribution.
- Initial values for the variance of the error terms $\left\{\sigma_i^2\right\}_i$. This choice mostly depends on the context and properties of the dataset. We suggest avoiding initial values with more than an order of magnitude different from the variability of data. In particular, a too high value might result in a model mostly capturing noise.

As a final note, let us stress that the EM algorithm depends on the initialisation and is only guaranteed to converge to local maxima of the likelihood function (McLachlan & Krishnan, 2007). Several strategies have been considered in the literature to tackle this issue such as simulated annealing (Ueda & Nakano, 1998) or repeated short runs (Biernacki et al., 2003). In this work, we chose the latter option.

3.3 Pseudocode

We wrap up this section with the pseudocode of the EM component of our complete algorithm, which we call Magma (standing for Multi tAsk Gaussian processes with common MeAn). The corresponding code is available at https://github.com/ArthurLeroy/MAGMA.

Algorithm 1 Magma: EM component

```
Initialise $m_0$ and $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$.
while not converged do
    E step: Compute the hyper-posterior distribution
            $p(\mu_0 \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}) = \mathcal{N}(\widehat{m}_0, \widehat{\mathbf{K}})$.

    M step: Estimate hyper-parameter by maximising
            $\widehat{\Theta} = \underset{\Theta}{\text{argmax}}\, \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i} \left[ p(\{\mathbf{y}_i\}_i, \mu_0 \mid \Theta) \right]$.
end while
return $\widehat{\Theta}$, $\widehat{m}_0$, $\widehat{\mathbf{K}}$.
```

3.4 Discussion of EM algorithms and alternatives

Let us stress that even though we focus on prediction purpose in this paper, the output of the EM algorithm already provides results on related FDA problems. The generative model in Yang et al. (2016) describes a Bayesian framework that resembles ours to smooth multiple curves simultaneously. However, modelling variance structure with an Inverse-Wishart process forces the use of an MCMC algorithm for inference or the introduction of a more tractable approximation in Yang et al. (2017). One can think of the learning through Magma and applying a single task GP regression on each individual as an *empirical Bayes* counterpart to their approach. Meanwhile, μ_0's hyper-posterior distribution also provides the probabilistic estimation of a mean curve from a set of functional data. The closest method to our approach can be found in Shi et al. (2007) and the following book Shi & Choi (2011). The authors also work in the context of a multi-task GP model, and one can retrieve the idea of defining a mean function μ_0 to overcome the weaknesses of classic GPs in making predictions far from observed data. However, since their model uses B-splines to estimate this mean function, the method only works if all individuals share the same grid of observations, and does not account for uncertainty over μ_0.

4 Prediction

Once the hyper-parameters of the model have been learned, we can focus on our main goal: prediction for new individuals at unobserved timestamps. Since $\widehat{\Theta}$ is known and for the sake of concision, we omit conditioning on $\widehat{\Theta}$ in the sequel. Note there are two cases for prediction (referred to as *Type I* and *Type II* in Shi & Cheng 2014, Section 3.2.1), depending on whether we observe some data or not for any new individual we wish to predict on. We denote by the index $*$ a new individual for whom we want to make a prediction, say at timestamps $\mathbf{t}^p$. If there are no available data for this individual, we have no $*$-specific information, and the prediction is merely given by $p(\mu_0(\mathbf{t}^p) \mid \{\mathbf{y}_i\}_i)$. This quantity may be considered as the 'generic' (or *Type II*) prediction according to the trained model, and only informs us through the mean process. Computing $p(\mu_0(\mathbf{t}^p) \mid \{\mathbf{y}_i\}_i)$ is also one of the steps leading to the prediction for a partially observed new individual (*Type I*). The latter being the most compelling case, we consider *Type II* prediction as a particular case of the full *Type I* procedure, described below.

If we observe $\left\{\mathbf{t}_*, y_*(\mathbf{t}_*)\right\}$ for the new individual, the multi-task GP prediction is obtained in our model by computing the posterior distribution $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \left\{\mathbf{y}_i\right\}_i)$. Note that the conditioning is taken over $y_*(\mathbf{t}_*)$, as for any GP regression, but also on $\left\{\mathbf{y}_i\right\}_i$, which is specific to our multi-task setting. The procedure for computing this distribution requires to successively complete the following steps:

1. choose a grid of prediction $\mathbf{t}^p$ and define the pooled vector of timestamps $\mathbf{t}_*^p$,
2. compute the hyper-posterior distribution of μ_0 at $\mathbf{t}_*^p$: $p(\mu_0(\mathbf{t}_*^p) \mid \left\{\mathbf{y}_i\right\}_i)$,
3. compute the multi-task prior distribution $p(y_*(\mathbf{t}_*^p) \mid \left\{\mathbf{y}_i\right\}_i)$,
4. compute hyper-parameters θ_* associated with the new individual (optional),
5. compute the multi-task posterior distribution: $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \left\{\mathbf{y}_i\right\}_i)$.

4.1 Posterior inference on the mean process

As mentioned above, we observed a new individual at timestamps $\mathbf{t}_*$. The GP regression consists in arbitrarily choosing a vector $\mathbf{t}^p$ of timestamps for which we aim at making predictions. Then, we define new notation for the pooled vector of timestamps $\mathbf{t}_*^p = \begin{bmatrix} \mathbf{t}^p \\ \mathbf{t}_* \end{bmatrix}$, which will serve as a working grid to define the prior and posterior distributions involved in the prediction process. One can note that, although not mandatory in theory, it is often a good idea to include the observed timestamps of training individuals, $\mathbf{t}$, within $\mathbf{t}_*^p$ since they match locations that contain information for the mean process to 'help' the prediction. In particular, if $\mathbf{t}_*^p = \mathbf{t}$, the computation of μ_0's hyper-posterior distribution is not necessary since $p(\mu_0(\mathbf{t}) \mid \left\{\mathbf{y}_i\right\}_i)$ has previously been obtained from the EM algorithm. However, in general, it is necessary to compute the hyper-posterior $p(\mu_0(\mathbf{t}_*^p) \mid \left\{\mathbf{y}_i\right\}_i)$ at the new timestamps. The idea remains similar to the E step aforementioned, and we obtain the following result.

Proposition 4 *Let $\mathbf{t}_*^p$ be a vector of timestamps of size $\tilde{N}$. The hyper-posterior distribution of μ_0 remains Gaussian*:

$$p\left(\mu_0(\mathbf{t}_*^p) \mid \left\{\mathbf{y}_i\right\}_i\right) = \mathcal{N}\left(\mu_0(\mathbf{t}_*^p); \widehat{m}_0(\mathbf{t}_*^p), \widehat{\mathbf{K}}_*^p\right),$$

with:

- $\widehat{\mathbf{K}}_*^p = \left(\tilde{\mathbf{K}}^{-1} + \sum\limits_{i=1}^{M} \tilde{\boldsymbol{\Psi}}_i^{-1}\right)^{-1}$,
- $\widehat{m}_0(\mathbf{t}_*^p) = \widehat{\mathbf{K}}_*^p \left(\tilde{\mathbf{K}}^{-1} m_0\left(\mathbf{t}_*^p\right) + \sum\limits_{i=1}^{M} \tilde{\boldsymbol{\Psi}}_i^{-1} \tilde{\mathbf{y}}_i\right)$,

where we used the shortening notation:

- $\tilde{\mathbf{K}} = k_{\widehat{\theta}_0}\left(\mathbf{t}_*^p, \mathbf{t}_*^p\right)$ ($\tilde{N} \times \tilde{N}$ matrix),
- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t)\right)_{t \in \mathbf{t}_*^p}$ ($\tilde{N}$-size vector),
- $\tilde{\boldsymbol{\Psi}}_i = \left[\mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\widehat{\theta}_i, \widehat{\sigma}_i^2}\left(t, t'\right)\right]_{t,t' \in \mathbf{t}_*^p}$ ($\tilde{N} \times \tilde{N}$ matrix).

Proof The sketch of the proof is similar to Proposition 1 in the E step. The only technicality consists in dealing carefully with the dimensions of vectors and matrices involved, and whenever relevant, to define augmented versions of $\mathbf{y}_i$ and $\boldsymbol{\Psi}_{\hat{\theta}_i, \hat{\sigma}_i^2}$ with 0 elements at unobserved timestamps' position for the i-th individual. Note that if we pick a vector $\mathbf{t}_*^p$ including only some of the timestamps from $\mathbf{t}_i$, information coming from y_i at the remaining timestamps is ignored. We defer details to Sect. 8.1. □

4.2 Computing the multi-task prior distribution

According to our generative model, given the mean process, any new individual $*$ is modelled as:

$$y_*(\cdot) \mid \mu_0(\cdot) \sim \mathcal{GP}\Big(\mu_0(\cdot), \boldsymbol{\Psi}_{\theta_*, \sigma_*^2}(\cdot, \cdot)\Big).$$

Therefore, for any finite-dimensional vector of timestamps, and in particular for $\mathbf{t}_*^p$, $p(y_*(\mathbf{t}_*^p) \mid \mu_0(\mathbf{t}_*^p))$ is a multivariate Gaussian. Moreover, from this distribution and μ_0's hyper-posterior, we can figure out the multi-task prior distribution over $y_*(\mathbf{t}_*^p)$, defined as below.

Proposition 5 *For any set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of y_* is given by*

$$p(y_*(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\Big(y_*(\mathbf{t}_*^p); \hat{m}_0(\mathbf{t}_*^p), \hat{\mathbf{K}}_*^p + \boldsymbol{\Psi}_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p}\Big). \quad (3)$$

Proof To compute this prior, we need to integrate out the mean process μ_0 in $p(y_* \mid \mu_0, \{\mathbf{y}_i\}_i)$, whereas the multi-task aspect remains through the conditioning over $\{\mathbf{y}_i\}_i$. We omit the writing of timestamps, by using the simplified notation μ_0 and y_* instead of $\mu_0(\mathbf{t}_*^p)$ and $y_*(\mathbf{t}_*^p)$, respectively. We first use the assumption that $\{y_i \mid \mu_0\}_{i \in \{1, \dots, M\}} \perp\!\!\!\perp y_* \mid \mu_0$, *i.e.*, the individuals are independent conditionally to μ_0. Then, one can notice that the two distributions involved within the integral are Gaussian, which leads to the explicit Gaussian target distribution after integration.

$$\begin{aligned} p(y_* \mid \{\mathbf{y}_i\}_i) &= \int p\big(y_*, \mu_0 \mid \{\mathbf{y}_i\}_i\big) \, \mathrm{d}\mu_0 \\ &= \int p\big(y_* \mid \mu_0, \{\mathbf{y}_i\}_i) p(\mu_0 \mid \{\mathbf{y}_i\}_i\big) \, \mathrm{d}\mu_0 \\ &= \int \underbrace{p\big(y_* \mid \mu_0)\big)}_{\mathcal{N}\left(y_*; \mu_0, \boldsymbol{\Psi}_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p}\right)} \underbrace{p(\mu_0 \mid \{\mathbf{y}_i\}_i\big)}_{\mathcal{N}\left(\mu_0; \hat{m}_0, \hat{\mathbf{K}}_*^p\right)} \mathrm{d}\mu_0. \end{aligned}$$

This convolution of two Gaussians remains Gaussian (Bishop, 2006, Chapter 2.3.3). The mean parameter is then given by

$$
\begin{aligned}
\mathbb{E}_{y_* \mid \{\mathbf{y}_i\}_i}\left[y_* \right] &= \int y_* \, p\left(y_* \mid \left\{\mathbf{y}_i\right\}_i\right) \mathrm{d}y_* \\
&= \int y_* \int p\left(y_* \mid \mu_0\right) p(\mu_0 \mid \left\{\mathbf{y}_i\right\}_i) \, \mathrm{d}\mu_0 \, \mathrm{d}y_* \\
&= \int \left(\int y_* p\left(y_* \mid \mu_0\right) \mathrm{d}y_* \right) p(\mu_0 \mid \left\{\mathbf{y}_i\right\}_i) \, \mathrm{d}\mu_0 \\
&= \int \mathbb{E}_{y_* \mid \mu_0}\left[y_* \right] p(\mu_0 \mid \left\{\mathbf{y}_i\right\}_i) \, \mathrm{d}\mu_0 \\
&= \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mathbb{E}_{y_* \mid \mu_0}\left[y_* \right] \right] \\
&= \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mu_0 \right] \\
&= \widehat{m}_0 .
\end{aligned}
$$

Following the same idea, the second-order moment is given by

$$
\begin{aligned}
\mathbb{E}_{y_* \mid \{\mathbf{y}_i\}_i}\left[y_*^2 \right] &= \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mathbb{E}_{y_* \mid \mu_0}\left[y_*^2 \right] \right] \\
&= \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mathbb{V}_{y_* \mid \mu_0}\left[y_* \right] + \mathbb{E}_{y_* \mid \mu_0}\left[y_* \right]^2 \right] \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mu_0^2 \right] \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \mathbb{V}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mu_0 \right] + \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i}\left[\mu_0 \right]^2 \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \widehat{\mathbf{K}} + \widehat{m}_0^2,
\end{aligned}
$$

hence

$$
\begin{aligned}
\mathbb{V}_{y_* \mid \{\mathbf{y}_i\}_i}\left[y_* \right] &= \mathbb{E}_{y_* \mid \{\mathbf{y}_i\}_i}\left[y_*^2 \right] - \mathbb{E}_{y_* \mid \{\mathbf{y}_i\}_i}\left[y_* \right]^2 \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \widehat{\mathbf{K}} + \widehat{m}_0^2 - \widehat{m}_0^2 \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \widehat{\mathbf{K}}.
\end{aligned}
$$

□

Note that the process $y_*(\cdot) \mid \left\{\mathbf{y}_i\right\}_i$ is not strictly a GP, although its finite-dimensional evaluation (3) remains Gaussian. The covariance structure cannot be expressed as a kernel that could be directly evaluated at any timestamps: the process is known as a *degenerated GP*. In practice however, this does not bear much consequence as any arbitrary vector of timestamps τ can be chosen at first, and computing hyper-posterior $p(\mu_0(\tau) \mid \left\{\mathbf{y}_i\right\}_i)$ still yields to the Gaussian distribution $p(y_*(\tau) \mid \left\{\mathbf{y}_i\right\}_i)$ as above. For the sake of simplicity, we now rename the covariance matrix of the multi-task prior distribution:

$$
\widehat{\mathbf{K}}_*^p + \boldsymbol{\Psi}_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p} = \boldsymbol{\Gamma}_*^p = \begin{pmatrix} \boldsymbol{\Gamma}_{pp} & \boldsymbol{\Gamma}_{p*} \\ \boldsymbol{\Gamma}_{*p} & \boldsymbol{\Gamma}_{**} \end{pmatrix},
$$

where the indices in the blocks of the matrix correspond to the associated timestamps $\mathbf{t}^p$ and $\mathbf{t}_*$.

4.3 Learning the new hyper-parameters

When we collect data points for a new individual, as in the single-task GPs setting, we would need to learn the hyper-parameters of its covariance kernel before making predictions. A salient fact in our multi-task approach is that we consider this step being part of the prediction process, for two main reasons. First, the model is already trained for individuals $i = 1, \dots, M$, and this training is independent of the future individual $*$ or the choice of prediction timestamps. Since learning these new hyper-parameters requires knowledge of $\mu(\mathbf{t}_*^p)$ and thus of the prediction timestamps, we cannot compute them beforehand. Second, learning these hyper-parameters with the *empirical Bayes* approach only requires maximisation of a Gaussian likelihood which is negligible in computing time compared to the previous EM algorithm. As for single-task GP, we have the following estimates for hyper-parameters:

$$\begin{aligned}\widehat{\Theta}_* &= \underset{\Theta_*}{\operatorname{argmax}}\, p(y_*(\mathbf{t}_*) \mid \{\mathbf{y}_i\}_i, \Theta_*) \\ &= \underset{\Theta_*}{\operatorname{argmax}}\, \mathcal{N}\big(y_*(\mathbf{t}_*); \widehat{m}_0(\mathbf{t}_*), \boldsymbol{\Gamma}_{**}^{\Theta_*}\big).\end{aligned}$$

Note that this step is optional depending on the modelling assumption: in the common hyper-parameters model (i.e. $(\theta, \sigma^2) = (\theta_i, \sigma_i^2), \forall i \in \mathcal{I}$), any new individual will also share the same hyper-parameters and we already have $\widehat{\Theta}_* = (\widehat{\theta}_*, \widehat{\sigma}_*^2) = (\widehat{\theta}, \widehat{\sigma}^2)$ from the EM algorithm.

4.4 Prediction

We can rewrite the multi-task prior distribution, by separating observed and prediction timestamps, as:

$$\begin{aligned}p(y_*(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i) &= p(y_*(\mathbf{t}^p), y_*(\mathbf{t}_*) \mid \{\mathbf{y}_i\}_i) \\ &= \mathcal{N}\big(y_*(\mathbf{t}_*^p); \widehat{m}_0(\mathbf{t}_*^p), \boldsymbol{\Gamma}_*^p\big) \\ &= \mathcal{N}\left(\begin{bmatrix} y_*(\mathbf{t}^p) \\ y_*(\mathbf{t}_*)\end{bmatrix}; \begin{bmatrix} \widehat{m}_0(\mathbf{t}^p) \\ \widehat{m}_0(\mathbf{t}_*)\end{bmatrix}, \begin{pmatrix} \boldsymbol{\Gamma}_{pp} & \boldsymbol{\Gamma}_{p*} \\ \boldsymbol{\Gamma}_{*p} & \boldsymbol{\Gamma}_{**}\end{pmatrix}\right).\end{aligned}$$

As usual, the conditional distribution remains Gaussian, and the multi-task posterior distribution is given by:

$$p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \mathcal{N}\Big(y_*(\mathbf{t}^p); \widehat{\mu}_0^p, \widehat{\boldsymbol{\Gamma}}^p\Big),$$

where:

- $\widehat{\mu}_0^p = \widehat{m}_0(\mathbf{t}^p) + \boldsymbol{\Gamma}_{p*}\boldsymbol{\Gamma}_{**}^{-1}\big(y_*(\mathbf{t}_*) - \widehat{m}_0(\mathbf{t}_*)\big)$,
- $\widehat{\boldsymbol{\Gamma}}^p = \boldsymbol{\Gamma}_{pp} - \boldsymbol{\Gamma}_{p*}\boldsymbol{\Gamma}_{**}^{-1}\boldsymbol{\Gamma}_{*p}$.

Although this predictive distribution presents a formulation nicely analogous to standard GPs, let us emphasise on the terms $\widehat{m}_0(\mathbf{t}_*^p)$ and $\boldsymbol{\Gamma}_*^p$, which embed crucial information from training individuals for the mean prediction to be more relevant even in far from the observed points $y_*(\mathbf{t}_*)$.

5 Complexity analysis for training and prediction

Computational complexity is of paramount importance in GPs as it quickly scales with large datasets. The classical cost to train a GP is $\mathcal{O}(N^3)$, and $\mathcal{O}(N^2)$ for prediction (Rasmussen & Williams, 2006) where N is the number of data points (although there exist various sparse approximations, see Sect. 7 for references). Moreover, multi-task GP models lying on LMC approaches typically present a complexity of $\mathcal{O}(M^3N^3)$ in training, which can be diminished when using sparse approximations (Álvarez and Lawrence 2011). As detailed below, our model reaches a reduction to $\mathcal{O}((M+1)N^3)$ for the training complexity in a similar context (common grid of timestamps for all individuals), without using any sparse approximation.

More specifically, since Magma uses information from M individuals, each of them providing N_i observations, these quantities determine the overall complexity of the algorithm. If we recall that N is the number of distinct timestamps (i.e. $N \leq \sum_{i=1}^{M} N_i$), the training complexity is $\mathcal{O}\big(M \times N_i^3 + N^3\big)$ (*i.e.* the complexity of each EM iteration). As usual with GPs, the cubic costs come from the inversion of the corresponding matrices, and here, the constant is proportional to the number of iterations of the EM algorithm. The dominating term in this expression depends on the values of M, relatively to N. For a large number of individuals with many common timestamps ($MN_i \gtrsim N$), the first term dominates. For diverse timestamps among individuals ($MN_i \lesssim N$), the second term becomes the primary burden, as in any GP problem. During the prediction step, the re-computation of μ_0's hyper-posterior implies the inversion of a $\tilde{N} \times \tilde{N}$ (dimension of $\mathbf{t}_*^p$) which has a $\mathcal{O}(\tilde{N}^3)$ complexity while the new hyper-parameters estimation's cost is $\mathcal{O}(N_*^3)$. In practice, the most computationally-expensive steps can be performed in advance to allow for quick on-the-fly prediction when collecting new data. If we observe the training dataset once and pre-compute the hyper-posterior of μ_0 on a fine grid on which to predict later, the immediate computational cost for each new individual is identical to the one of the single-task GP regression.

6 Experimental results

We evaluate our Magma algorithm on synthetic data and two real datasets. The classical GP regression on single tasks separately is used as the baseline alternative for predictions. While it is not expected to perform well on the dataset used, the comparison highlights the interest of multi-task approaches. To our knowledge, the only alternative to Magma is the GPFDA algorithm from Shi et al. (2007), Shi & Choi (2011), described in Sect. 3.4, and the associated R package *GPFDA*, which is applied during the experiments. Throughout the section, the standard *Exponentiated Quadratic* kernel (see Eq. (1)) is used both for simulating the data and for modelling the covariance structures in the three algorithms. Hence, each kernel is associated with $\theta = \{v, \ell\}$, $v, \ell \in \mathbb{R}^+$, a set of variance and length-scale hyper-parameters, respectively. Each simulated dataset has been drawn from the sampling scheme below:

1. Draw a random working grid $\mathbf{t} \subset [\,0, 10\,]$ of $N = 200$ timestamps, and a number M of individuals.

2. Define a prior mean function : $m_0(t) = at + b, \ \forall t \in \mathbf{t}$, where $a \in [\,-2, 2\,]$ and $b \in [\,0, 10\,]$ are drawn uniformly.
3. Draw hyper-parameters uniformly for μ_0's kernel : $\theta_0 = \{v_0, \ell_0\}$, where $v_0 \in \left[\,1, \exp(5)\,\right]$ and $\ell_0 \in \left[\,1, \exp(2)\,\right]$.
4. Draw $\mu_0(\mathbf{t}) \sim \mathcal{N}\left(m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right)$.
5. $\forall i \in \mathcal{I}$, draw $v_i \in \left[\,1, \exp(5)\,\right]$, $\ell_i \in \left[\,1, \exp(2)\,\right]$, and $\sigma_i^2 \in [\,0, 1\,]$ uniformly.
6. $\forall i \in \mathcal{I}$, draw a subset $\mathbf{t}_i \subset \mathbf{t}$ of $N_i = 30$ timestamps uniformly, and draw $\mathbf{y}_i \sim \mathcal{N}\left(\mu_0(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}\right)$.

This procedure provides a synthetic dataset $\left\{\mathbf{t}_i, \mathbf{y}_i\right\}_i$, and its associated mean process $\mu_0(\mathbf{t})$. Those quantities are used to train the model, make predictions with each algorithm, and then compute errors in μ_0 estimation and forecasts. We recall that the Magma algorithm enables two different settings depending on the model's assumption over hyper-parameters (HP), and we refer to them as *Common HP* and *Different HP* in the following. In order to test these two contexts, differentiated datasets have been generated, by drawing *Common HP data* or *Different HP data* for each individual at step 5. We previously presented the idea of the model used in GPFDA, and, although the algorithm has many features (in particular about the type and number of input variables), it is not yet usable when timestamps are different among individuals. Therefore, two frameworks are considered, *Common grid* and *Uncommon grid*, to take this specification into account. Thus, the comparison between the different methods can only be performed on data generated under the settings *Common HP* and *Common grid*, and the effect of those different settings on Magma is analysed separately. Moreover, the initialisation for the prior mean function, $m_0(\cdot)$, is set to be constant, equal to 0 for each algorithm. Except in some experiments, where the influence of the number of individuals is analysed, the generic value is $M = 20$. In the case of prediction on unobserved timestamps for a new individual, the first 20 data points are used as observations, and the remaining 10 are taken as test values. Optimisation of the hyper-parameters is performed by likelihood maximisation, using the L-BFGS-B algorithm (Morales & Nocedal, 2011; Nocedal, 1980) in all methods. The convergence criterion for all algorithms is reached if the difference of log-likelihood between two iterations is lower than 10^{-2}. In general, the EM algorithm in Magma converges in a few iterations, typically fewer than 5 with the *Common HP* setting, and rarely more than 15 even with the *Different HP* setting.

6.1 Illustration on a simple example

To illustrate the multi-task approach of Magma, Fig. 2 displays a comparison between standard GP regression and Magma on a simple example, from a dataset simulated according to the scheme above and using the *Uncommon grid/Common HP* setting. Given the observed data (in black), values on a thin grid of unobserved timestamps are predicted and compared, in particular, with the true test values (in red). As expected, the GP regression provides a good fit close to the data points and then dives rapidly to the prior 0 with increasing uncertainty. Conversely, although the initialisation for the prior mean is 0 in Magma as well, the hyper-posterior distribution of μ_0 (dashed line) is estimated thanks to all individuals in the training dataset. This process acts as an informed prior helping GP prediction for the new individual, even far from its own observations. More precisely, 3 phases can be distinguished according to the level of information coming from the data: in the first one, close to the observed data ($t \in [\,1, 7\,]$), the two processes behave similarly, except for a slight increase in the variance for Magma, which is logical since the prediction also takes

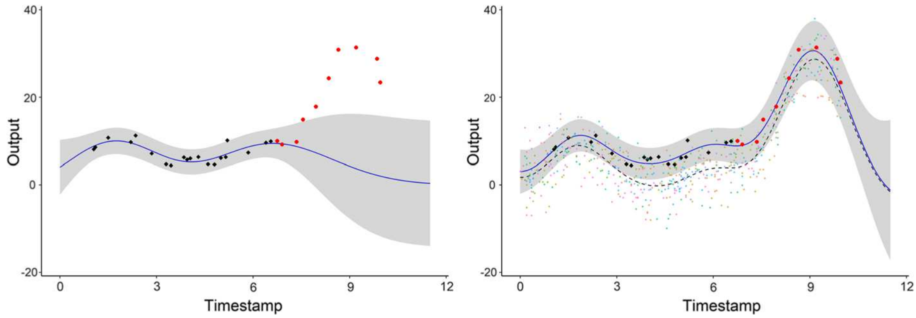

Fig. 2 Prediction curves (blue) of a new individual with associated 95% credible intervals (grey) for GP regression (left) and Magma (right). The dashed line represents the mean function $\widehat{m}_0$, from the hyper-posterior $p(\mu_0 \mid \{\mathbf{y}_i\}_i)$. Observed data points are in black, and testing data points are in red. The colourful backward points are the observations from the training dataset, each colour corresponding to a different individual (Color figure online)

uncertainty over μ_0 into account (see Eq. (3)); in the second one, on intervals of unobserved timestamps containing data points from the training dataset ($t \in [\,0, 1\,] \cup [\,7, 10\,]$), the prediction is guided by the information coming from other individuals through μ_0. In this context, the mean trajectory remains coherent and the uncertainty increases only slightly. In the third phase, where no observations are available, neither from the new individual nor from the training dataset ($t \in [\,10, 12\,]$), the prediction behaves as expected, with a slow drifting to the prior mean 0, with highly increasing variance. Overall, the multi-task framework provides reliable probabilistic predictions on a wider range of timestamps, potentially outside of the usual scope of GPs.

6.2 Performance comparison on simulated datasets

We confront the performance of Magma to alternatives in several situations and for different datasets. In the first place, the classical GP regression (GP), GPFDA and Magma are compared through their performance in prediction and estimation of the true mean process μ_0. In the prediction context, the performances are evaluated according to the following indicators:

- the mean squared error (MSE) which compares the predicted values to the true test values of the 10 last timestamps:

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

- the CI_{95} coverage (CIC_{95}), i.e. the percentage of unobserved data points effectively lying within the 95% credible interval defined from the predictive posterior distribution $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$:

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

Table 1 Average MSE (standard deviation) and average CIC_{95} (standard deviation) on 100 runs for GP, GPFDA and Magma

	Prediction		Estimation μ_0	
	MSE	CIC_{95}	MSE	CIC_{95}
Magma	18.7 (31.4)	93.8 (13.5)	1.3 (2)	94.3 (11.3)
GPFDA	31.8 (49.4)	90.4 (18.1)	2.4 (3.6)	⋆
GP	87.5 (151.9)	74.0 (32.7)		

⋆ : 99.6 (2.8), the measure of incertitude from the GPFDA package is not a genuine credible interval

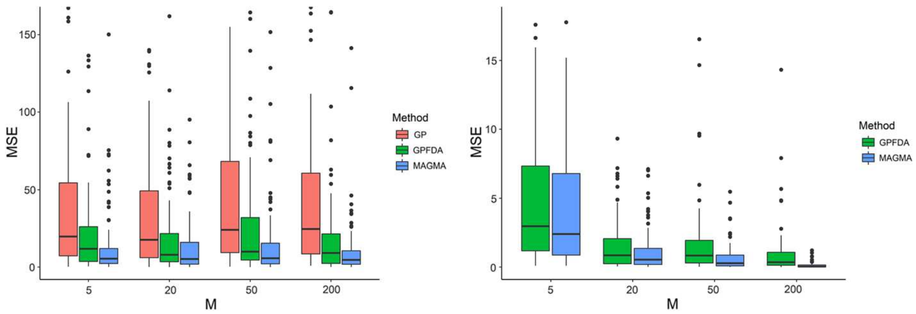

Fig. 3 MSE with respect to the number M of training individuals (boxplots are displayed from 100 runs in each case). *Left* prediction error on 10 testing points. *Right* estimation error of the true mean process μ_0

The CIC_{95} provides insights on the reliability of the predictive variance and should be as close to the value 95% as possible. Other values would indicate a tendency to underestimate or overestimate the uncertainty. Let us recall that GPFDA uses B-splines to estimate the mean process and does not account for uncertainty, contrarily to a probabilistic framework as Magma. However, a measure of uncertainty based on an empirical variance estimated from training curves is proposed (see Shi & Cheng, 2014, Section 3.2.1). In practice, this measure constantly overestimates the true variance, and their 95% empirical interval coverage is generally equal or close to 100%.

In the estimation context, the performances are evaluated thanks to another MSE, which compares the estimations to the true values of μ_0 at all timestamps:

$$\frac{1}{M} \sum_{i=1}^{M} \frac{1}{N_i} \sum_{k=1}^{N_i} \left(\mu_0^{\text{pred}}(t_i^k) - \mu_0^{\text{true}}(t_i^k) \right)^2 .$$

Table 1 presents the results obtained over 100 datasets, where the models are trained on $M = 20$ individuals, each of them observed on $N = 30$ common timestamps. As expected, both multi-task methods lead to better results than GP. However, Magma outperforms GPFDA, both in the estimation of μ_0 and in predictive performance. In terms of error as well as in uncertainty quantification, Magma provides more accurate results, in particular with a CI_{95} coverage close to the 95% expected value. Each method presents a quite high standard deviation for MSE in prediction, which is due to some datasets with particularly difficult values to predict, although most of the cases lead to small errors. This behaviour is reasonably expected since such 10-timestamps-ahead forecasts might sometimes be tricky. It can also be noticed on Fig. 3 that Magma consistently provides lower errors as well as less pathological behaviour, as it may sometimes occur with the B-splines modelling used in GPFDA.

To highlight the effect of the number of individuals M on the performance, Fig. 3 provides the same 100 runs trial as previously, for different values of M. The boxplots exhibit, for each method, the behaviour of the prediction and estimation MSE as information is added in the training dataset. Let us mention the absence of discernible changes as soon as $M > 200$. As expected, we notice on the right panel that adding information from new individuals improves the estimation of μ_0, leading to shallow errors for high values of M, in particular for Magma. Meanwhile, the left panel exhibits reasonably unchanged prediction performance with respect to the values of M, excepted for some random fluctuations. This property is expected for GP regression since no external information is used from the training dataset in this context. For both multi-tasks algorithms though, the estimation of μ_0 improves the prediction by one order of magnitude below the typical errors, even with only a few training individuals. Furthermore, since a new individual behaves independently through f_*, it is natural for a 10-points-ahead forecast to present intrinsic variations, despite an adequate estimation of the shared mean process.

To illustrate the advantage of multi-task methods, even for $M = 20$, we display on Fig. 4 the evolution of MSE according to the number of timestamps N that are assumed to be observed for the new individual on which we make predictions. These predictions remain computed on the last 10 timestamps, although in this experiment, we only observe the first 5, 10, 15, or 20 timestamps, in order to change the volume of information and the distance from training observations to targets. We observe on Fig. 4 that, as expected in a GP framework, the closer observations are to targets, the better the results. However, for multi-tasks approaches and in particular for Magma, the prediction remains consistently adequate even with few observations. Once more, sharing information across individuals significantly helps the prediction, even for small values of M or few observed data.

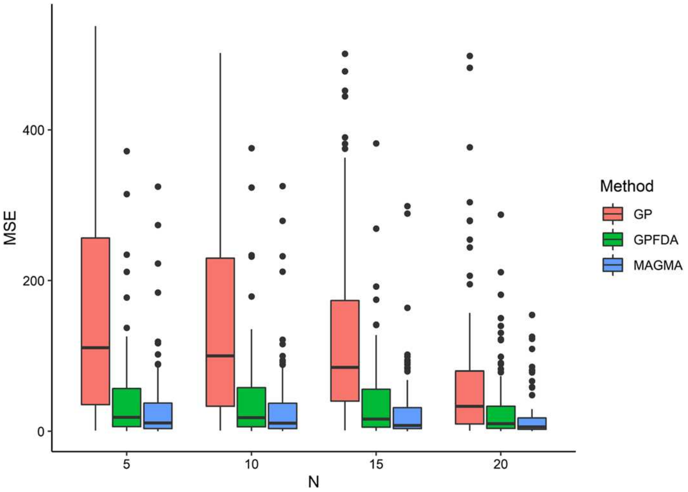

Fig. 4 MSE prediction error on the 10 last testing points with respect to the increasing number N of observed timestamps, among the first 20 points (boxplots are displayed from 100 runs in each case)

6.3 Magma's specific settings

As we previously discussed, different settings are available for Magma according to the nature of data and the model hypotheses. First, the *Common grid* setting corresponds to cases where all individuals share the same timestamps, whereas *Uncommon grid* is used otherwise. Moreover, Magma enables to consider identical hyper-parameters for all individuals or specific ones, as previously discussed in Sect. 2.2. To evaluate the effect of the different settings, performances in prediction and μ_0's estimation are evaluated in the following cases in Table 2:

- *Common HP*, when data are simulated with a common set of hyper-parameters for all individuals, and Proposition 3 is used for inference in Magma,
- *Different HP*, when data are simulated with its own set of hyper-parameters for each individual, and Proposition 2 is used for inference in Magma,
- *Common HP on different HP data*, when data are simulated with its own set of hyper-parameters for each individual, and Proposition 3 is used for inference in Magma.

Note that the first line of the table (*Common grid / Common HP*) of Table 2 is identical to the corresponding results in Table 1, providing reference values, significantly better than for other methods. The results obtained in Table 2 indicate that the Magma performance is not significantly altered by the settings used or the nature of the simulated data. To confirm the robustness of the method, the setting *Common HP* was applied to data generated by drawing different values of hyper-parameters for each individual (*Different HP data*). In this case, performances in prediction and estimation of μ_0 are slightly deteriorated, although Magma still provides quite reliable forecasts. This experience also highlights a particularity of the *Different HP* setting: looking at the estimation of μ_0 performance, we observe a significant decrease in the CI_{95} coverage, due to numerical instability in some pathological cases. Numerical issues, in particular during matrix inversions, are classical problems in the GP literature and, because of the potentially large number of different hyper-parameters to train, the probability for at least one of them to lead to a nearly singular matrix increases. In this case, one individual might overwhelm others in the calculus of μ_0's hyper-posterior (see Proposition 4), and thus lead to an underestimated posterior variance. This problem does not occur in the *Common HP* settings, since sharing the same hyper-parameters prevents the associated covariance matrices from running over

Table 2 Average MSE (standard deviation) and average CIC_{95} (standard deviation) on 100 runs for the different settings of Magma

		Prediction		Estimation of μ_0	
		MSE	CIC_{95}	MSE	CIC_{95}
Common HP	Common grid	18.7 (31.4)	93.8 (13.5)	1.3 (2)	94.3 (11.3)
	Uncommon grid	19.2 (43)	94.6 (13.1)	2.9 (2.6)	93.6 (9.2)
Different HP	Common grid	19.9 (54.7)	91.6 (17.8)	0.5 (0.4)	70.8 (24.3)
	Uncommon grid	14.5 (22.4)	89.1 (17.9)	2.5 (4.5)	81.1 (15.9)
Common HP on different HP data	Common grid	21.7 (36)	91 (19.8)	1.5 (1.2)	91.1 (13)
	Uncommon grid	18.1 (33)	92.5 (15.9)	3.2 (4.5)	93.4 (9.8)

Table 3 Average (standard deviation) training time (in s) for Magma and GPFDA on 100 runs for different numbers *M* of individuals in the training dataset

$M =$	5	10	50	100
Magma	5.2 (2.7)	7.6 (3.2)	24.2 (11.1)	42.8 (10)
GPFDA	1 (0.3)	2.1 (0.6)	10.7 (2.4)	23.1 (5.3)
Ratio	5.2	3.6	2.3	1.9

The relative running time between Magma and GPFDA is provided on the line *Ratio*

Table 4 Average (standard deviation) training and prediction time (in s) on 100 runs for different settings of Magma

		Train	Predict
Common HP	Common grid	12.6 (3.5)	0.1 (0)
	Uncommon grid	16.5 (11.4)	0.2 (0.1)
Different HP	Common grid	42.6 (20.5)	0.6 (0.1)
	Uncommon grid	40.2 (17)	0.6 (0.1)

each other. Thus, except if one specifically wants to smooth multiple curves presenting really different behaviours, keeping *Common HP* as a default setting appears as a reasonable choice. Let us notice that the estimation of μ_0 is slightly better for common than for uncommon grid since the estimation problem on the union of different timestamps is generally more difficult. However, this feature only depends on the nature of data.

6.4 Running times comparisons

The counterpart of the more accurate and general results provided by Magma is a natural increase in running time. Table 3 exhibits the raw and relative training times for GPFDA and Magma (prediction times are negligible and comparable in both cases), on data coming from the simulation scheme with varying values of M on a *Common grid* of $N = 30$ timestamps. The algorithms were run under the *3.6.1 R version*, on a laptop with a dual-core processor cadenced at 2.90GHz and an 8GB RAM. The reported computing times are in seconds, and for small to moderate datasets ($N \simeq 10^3$, $M \simeq 10^4$) the procedures ran in few minutes to few hours. The difference between the two algorithms is due to GPFDA modelling μ_0 as a deterministic function through B-splines smoothing, whereas Magma accounts for uncertainty. The ratio of computing times between the two methods tends to decrease as M increases, and stabilises around 2 for higher numbers of training individuals. This behaviour comes from the E step in Magma, which is incompressible and quite insensitive to the value of M. Roughly speaking, one needs to pay twice the computing price of GPFDA for Magma to provide (significantly) more accurate predictions and uncertainty over μ_0. Table 4 provides running times of Magma according to its different settings, with $M = 20$. Because the complexity is linear in M in each case, the ratio in running times would remain roughly similar no matter the value of M. Prediction time appears negligible compared to training time, and generally takes less than one second to run. Besides, the *Different HP* setting increases the running time since in this context M maximisations (instead of one for *Common HP*) are required at each EM iteration. In this case, the prediction also takes slightly longer because of the necessity to optimise hyper-parameters for the new individual. Although the nature of the grid of timestamps does not matter in itself, a key limitation lies in the dimension N of the pooled set of timestamps, which tends to get bigger when individuals have different timestamps from one another.

6.5 Application of Magma on swimmers' progression curves

Data and problematic

We consider the problem of performance prediction in competition for french swimmers. The French Swimming Federation provided us with an anonymised dataset, compiling the age and results of its members between 2000 and 2016. For each competitor, the race times are registered for competitions of 100m freestyle (50m swimming-pool). The database contains results from 1731 women and 7876 men, 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 ith swimmer is considered as the input variable (timestamp t) and the performance (in s) on a 100 m freestyle as the output ($y_i(t)$). For reasons of confidentiality and property, the raw dataset cannot be published. The analysis focuses on the youth period, from 10 to 20 years, where the progression is the most noticeable. In order to get relevant time series, we retained only individuals having a sufficient number of data points ($N_i \geq 5$) on the considered time period. For a young swimmer, observed during its first years of competition, we aim at modelling its progression curve and make predictions on its future performance in the subsequent years. Since we consider a decision-making problem involving irregular time series, the GP probabilistic framework is a natural choice to work on. Thereby, assuming that each swimmer in the database is a realisation y_i defined as previously, we expect Magma to provide multi-task predictions for a new young swimmer, that will benefit from information of other swimmers already observed at older ages. To study such modelling, and validate its efficiency in practice, we split the individuals into training and testing datasets with respective sizes:

- $M^F_{\text{train}} = 1039$, for the female training set,
- $M^F_{\text{test}} = 692$, for the female testing set,
- $M^M_{\text{train}} = 4726$, for the male training set,
- $M^M_{\text{test}} = 3150$, for the male testing set.

Inference on the hyper-parameters is performed thanks to the training dataset in both cases. Considering the different timestamps and the relative monotony of the progression curves, the settings *Uncommon grid/Common HP* has been used for Magma. The overall training lasted around 2 h with the same hardware configuration as for simulations. To compute MSE and the CI_{95} coverage, the data points of each individual in the testing set has been split into *observed* and *testing* timestamps. Since each individual has a different number of data points, the first 80% of timestamps are taken as *observed*, while the remaining 20% are considered as *testing* timestamps. Magma's predictions are compared with the true values of y_i at testing timestamps. As previously, both GP and Magma have been initialised with a constant 0 mean function. Initial values for hyper-parameters are also similar for all i, $\theta_0^{\text{ini}} = \theta_i^{\text{ini}} = (\exp(1), \exp(1))$ and $\sigma_i^{\text{ini}} = 0.4$. Those values are the default in Magma and remain adequate in the context of these datasets.

Results and interpretation The overall performance and comparison are summarised in Table 5.

Table 5 Average MSE (standard deviation) and average CIC_{95} (standard deviation) for prediction on french swimmer testing datasets

		MSE	CIC_{95}
Women	Magma	3.8 (10.3)	95.3 (15.9)
	GP	25.3 (97.6)	72.7 (37.1)
Men	Magma	3.7 (5.3)	93.9 (15.3)
	GP	22.1 (94.3)	78.2 (30.4)

We observe that Magma still provides excellent results in this context, and naturally outperforms predictions provided by a standard GP regression. As the progression curves present relatively monotonic variations and thus avoid pathological behaviours that could occur with synthetic data, the MSE in prediction remains very low. The CI_{95} coverage sticks close to the 95% expected value for Magma, indicating an adequate quantification of uncertainty. To illustrate these results, an example is displayed on Fig. 5 for both men and women. For a randomly chosen testing individual, we plot its predicted progression curve (in blue), where its first 15 data points are used as observations (in black), while the remaining true data points (in red) are displayed for comparison purpose. As previously observed in the simulation study, the simple GP quickly drifts to the prior 0 mean, as soon as data lack. However, for both men and women, the Magma predictions remain close to the true data, which also lie within the 95% credible interval. Even for long term forecast, where the mean prediction curve tends to overlap the mean process (dashed line), the true data remain in our range of uncertainty, as the credible interval widens far from observations. For clarity, we displayed only a few individuals from the training dataset

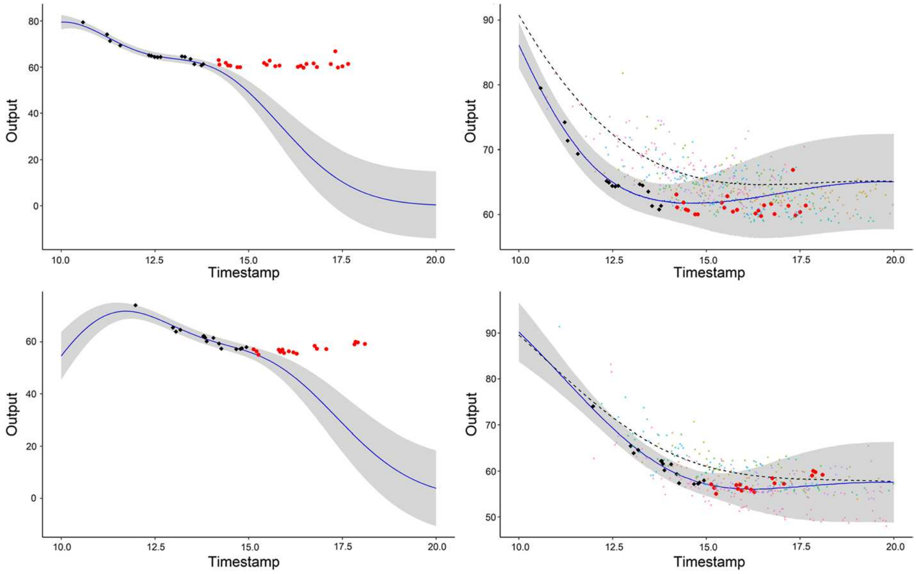

Fig. 5 Prediction curves (blue) for a testing individual with associated 95% credible intervals (grey) for GP regression (left) and Magma (right), for both women (top) and men (bottom). The dashed lines represent the mean functions $\widehat{m}_0$, from the hyper-posterior $p(\mu_0 \mid \{\mathbf{y}_i\}_i)$. Observed data points are in black, and testing data points are in red. The colourful backward points are observations from the training datasets, each colour corresponding to a different individual (Color figure online)

(colourful points) in the background. The mean process (dashed line) seems to represent the main trend of progression among swimmers correctly, even though we cannot numerically compare μ_0 to any real-life analogous quantity. From a more sport-related perspective, we can note that both genders present similar patterns of progression. However, while performances are roughly similar in mean trend before the age of 14, they start to differentiate afterwards and then converge to average times with approximatively a 5 s gap. Interestingly, the difference between men and women in terms of world records in swimming competitions for the 100m freestyle is currently 4.8 s (46.91 versus 51.71). These results, obtained under reasonable hypotheses on several hundreds of swimmers, seem to indicate that Magma would give quite reliable predictions for a new young swimmer. Furthermore, the uncertainty provided through the predictive posterior distribution offers an adequate degree of caution in a decision-making process.

7 Discussion

We have introduced a unified multi-task framework integrating a mean Gaussian process prior in the context of GP regression. While we believe that this process is an interesting object in itself, it also allows individuals to borrow information from each other and provides more accurate predictions, even far from data points. Furthermore, our method accounts for uncertainty in the mean process and remains applicable no matter eventual irregular timestamps in the grid of observations. The proposed algorithm, Magma, also presents a reduced computational complexity compared to previous multi-task GPs frameworks. Both on simulated and real-life datasets, we exhibited the efficiency of such an approach and studied some of its properties and possible settings. Magma outperforms the alternatives in estimation of the mean process as well as in prediction, and leads to a reliable quantification of uncertainty. We also displayed evidence of its predictive efficiency for real-life problems and provided some insights on practical interpretations about the mean process.

Despite the extensive literature on these aspects of GPs, our model does not yet include sparse approximations. While these aspects remain beyond the scope of the present paper, we might aim at adapting existing approaches (Snelson & Ghahramani, 2006; Quiñonero-Candela et al., 2007; Titsias, 2009) in our model to widen its applicability. Another possible avenue is an adaptation to the classification context, which is presented in Rasmussen and Williams (2006, Chapter 3). Besides, this work leaves the door open to improvement as we tackled here the problem of unidimensional regression: enabling either multidimensional or mixed type of inputs as in Shi & Choi (2011) would be of interest. To conclude, the hypothesis of a unique underlying mean process might be considered too restrictive for some datasets, and enabling cluster-specific mean processes would be a relevant extension.

8 Proofs

Note that the proof of Proposition 1 is a particular case of the proof below, where $\boldsymbol{\tau} = \mathbf{t}$ exactly (where $\boldsymbol{\tau}$ is the set of timestamps the hyper-posterior is to be computed on). Moreover, in order to keep an analytical expression for μ_0's hyper-posterior distribution, we discard the superfluous information contained in $\left\{\mathbf{y}_i\right\}_i$ at timestamps on which the hyper-posterior is not to be computed. Hence, the proof below states that the remaining data points are observed on subsets $\left\{\boldsymbol{\tau}_i\right\}_i$ of $\boldsymbol{\tau}$.

8.1 Proof of Proposition 4

Let $\boldsymbol{\tau}$ be a finite vector of timestamps, and $\left\{\boldsymbol{\tau}_i\right\}_i$ such as $\forall i \in \mathcal{I},\ \ \boldsymbol{\tau}_i \subset \boldsymbol{\tau}$. We define convenient notation:

- $\boldsymbol{\mu}_0^{\boldsymbol{\tau}} = \mu_0(\boldsymbol{\tau})$,
- $\mathbf{m}_0^{\boldsymbol{\tau}} = m_0(\boldsymbol{\tau})$,
- $\boldsymbol{\mu}_0^{\boldsymbol{\tau}_i} = \mu_0(\boldsymbol{\tau}_i),\ \forall i \in \mathcal{I}$,
- $\mathbf{y}_i^{\boldsymbol{\tau}_i} = y_i(\boldsymbol{\tau}_i),\ \forall i \in \mathcal{I}$,
- $\boldsymbol{\Psi}_i = \psi_{\theta_i,\sigma_i^2}(\boldsymbol{\tau}_i, \boldsymbol{\tau}_i), \forall i \in \mathcal{I}$,
- $\mathbf{K} = k_{\theta_0}(\boldsymbol{\tau}, \boldsymbol{\tau})$.

Moreover, for a covariance matrix C, and $u, v \in \boldsymbol{\tau}$, we note $[\,C\,]_{uv}^{-1}$ the element of the inverse matrix at row associated with timestamp u, and column associated with timestamp v. We also ignore the conditionings over $\widehat{\Theta}$, $\boldsymbol{\tau}_i$ and $\boldsymbol{\tau}$ to maintain simple expressions. By construction of the models, we have:

$$\begin{aligned} p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} \mid \left\{\mathbf{y}_i^{\boldsymbol{\tau}_i}\right\}_i) &\propto p(\left\{\mathbf{y}_i^{\boldsymbol{\tau}_i}\right\}_i \mid \boldsymbol{\mu}_0^{\boldsymbol{\tau}}) p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}) \\ &\propto \left\{ \prod_{i=1}^{M} p(\mathbf{y}_i^{\boldsymbol{\tau}_i} \mid \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}) \right\} p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}) \\ &\propto \left\{ \prod_{i=1}^{M} \mathcal{N}\big(\mathbf{y}_i^{\boldsymbol{\tau}_i}; \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}, \boldsymbol{\Psi}_i)\big) \right\} \mathcal{N}\big(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}; \mathbf{m}_0^{\boldsymbol{\tau}}, \mathbf{K}\big). \end{aligned}$$

The term $\mathcal{L}_1 = -(1/2) \log p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} \mid \left\{\mathbf{y}_i^{\boldsymbol{\tau}_i}\right\}_i)$ associated with the hyper-posterior remains quadratic and we may find the corresponding Gaussian parameters by identification:

$$\begin{aligned} \mathcal{L}_1 &= \sum_{i=1}^{M} \left\{ \left(\mathbf{y}_i^{\boldsymbol{\tau}_i} - \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}\right)^{\intercal} \boldsymbol{\Psi}_i^{-1} \left(\mathbf{y}_i^{\boldsymbol{\tau}_i} - \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}\right) + C_i \right\} + \left(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} - \mathbf{m}_0^{\boldsymbol{\tau}}\right)^{\intercal} \mathbf{K}^{-1} \left(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} - \mathbf{m}_0^{\boldsymbol{\tau}}\right) + C_0 \\ &= \boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal} \mathbf{K}^{-1} \boldsymbol{\mu}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M} \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i \intercal} \boldsymbol{\Psi}_i^{-1} \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i} - 2 \left(\boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal} \mathbf{K}^{-1} \mathbf{m}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M} \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i \intercal} \boldsymbol{\Psi}_i^{-1} \mathbf{y}_i^{\boldsymbol{\tau}_i} \right) + C \\ &= \sum_{u \in \boldsymbol{\tau}} \sum_{v \in \boldsymbol{\tau}} \mu_0(u) [\,\mathbf{K}\,]_{uv}^{-1} \mu_0(v) + \sum_{i=1}^{M} \sum_{u \in \boldsymbol{\tau}_i} \sum_{v \in \boldsymbol{\tau}_i} \mu_0(u) \left[\boldsymbol{\Psi}_i\right]_{uv}^{-1} \mu_0(v) \\ &\quad - 2 \sum_{u \in \boldsymbol{\tau}} \sum_{v \in \boldsymbol{\tau}} \mu_0(u) [\,\mathbf{K}\,]_{uv}^{-1} m_0(v) - 2 \sum_{i=1}^{M} \sum_{u \in \boldsymbol{\tau}_i} \sum_{v \in \boldsymbol{\tau}_i} \mu_0(u) \left[\boldsymbol{\Psi}_i\right]_{uv}^{-1} y_i(v) + C, \end{aligned}$$

where we entirely decomposed the vector-matrix products. We factorise the expression according to the common timestamps between τ_i and τ. Since for all $i, \boldsymbol{\tau}_i \subset \boldsymbol{\tau}$, let us introduce a dummy indicator function $\mathbb{1}_{\boldsymbol{\tau}_i} = \mathbb{1}_{\{u,v \in \boldsymbol{\tau}_i\}}$ to write:

$$\begin{aligned}\sum_{i=1}^{M}\sum_{u\in\boldsymbol{\tau}_i}\sum_{v\in\boldsymbol{\tau}_i} A(u,v) &= \sum_{i=1}^{M}\sum_{u\in\boldsymbol{\tau}}\sum_{v\in\boldsymbol{\tau}} \mathbb{1}_{\boldsymbol{\tau}_i} A(u,v)\\ &= \sum_{u\in\boldsymbol{\tau}}\sum_{v\in\boldsymbol{\tau}}\sum_{i=1}^{M} \mathbb{1}_{\boldsymbol{\tau}_i} A(u,v),\end{aligned}$$

subsequently, we can gather the sums such as:

$$\begin{aligned}\mathcal{L}_1 &= \sum_{u\in\boldsymbol{\tau}}\sum_{v\in\boldsymbol{\tau}}\Bigg(\mu_0(u)[\,\mathbf{K}\,]_{uv}^{-1}\mu_0(v) + \sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\mu_0(u)\big[\boldsymbol{\Psi}_i\big]_{uv}^{-1}\mu_0(v)\\ &\quad - 2\mu_0(u)[\,\mathbf{K}\,]_{uv}^{-1}m_0(v) - 2\sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\mu_0(u)\big[\boldsymbol{\Psi}_i\big]_{uv}^{-1}y_i(v)\Bigg) + C\\ &= \sum_{u\in\boldsymbol{\tau}}\sum_{v\in\boldsymbol{\tau}}\Bigg(\mu_0(u)\Big([\,\mathbf{K}\,]_{uv}^{-1} + \sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\big[\boldsymbol{\Psi}_i\big]_{uv}^{-1}\Big)\mu_0(v)\\ &\quad - 2\mu_0(u)\Big([\,\mathbf{K}\,]_{uv}^{-1}m_0(v) + \sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\big[\boldsymbol{\Psi}_i\big]_{uv}^{-1}y_i(v)\Big)\Bigg) + C\\ &= \boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal}\Big(\mathbf{K}^{-1} + \sum_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\Big)\boldsymbol{\mu}_0^{\boldsymbol{\tau}} - 2\boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal}\Big(\mathbf{K}^{-1}\mathbf{m}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i^{\boldsymbol{\tau}}\Big) + C,\end{aligned}$$

where the $\mathbf{y}_i$ and $\boldsymbol{\Psi}_i$ are completed by zeros:

- $\tilde{\mathbf{y}}_i^{\boldsymbol{\tau}} = \mathbb{1}_{\boldsymbol{\tau}_i} y_i(\boldsymbol{\tau})$,
- $\big[\tilde{\boldsymbol{\Psi}}_i\big]_{uv}^{-1} = \mathbb{1}_{\boldsymbol{\tau}_i}\big[\boldsymbol{\Psi}_i\big]_{uv}^{-1}, \ \forall u, v \in \boldsymbol{\tau}$.

By identification of the quadratic form, we reach:

$$p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} \mid \{\mathbf{y}_i^{\boldsymbol{\tau}_i}\}_i) = \mathcal{N}\Big(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}; \widehat{m}_0(\boldsymbol{\tau}), \widehat{\mathbf{K}}\Big),$$

with,

- $\widehat{\mathbf{K}} = \left(\mathbf{K}^{-1} + \sum\limits_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\right)^{-1}$,
- $\widehat{m}_0(\boldsymbol{\tau}) = \widehat{\mathbf{K}}\left(\mathbf{K}^{-1}\mathbf{m}_0^{\boldsymbol{\tau}} + \sum\limits_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i^{\boldsymbol{\tau}}\right)$.

□

8.2 Proof of Propositions 2 and 3

Since the central part of the proofs is similar for both propositions, we detail the calculus by denoting $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$ for generality, and dissociating the two cases only when necessary. Before considering the maximisation, we notice that the joint density can be developed as:

$$\begin{aligned} p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta) &= p(\{\mathbf{y}_i\}_i \mid \mu_0(\mathbf{t}), \Theta)\, p(\mu_0(\mathbf{t}) \mid \Theta) \\ &= \prod_{i=1}^{M} \left\{p(\mathbf{y}_i \mid \mu_0(\mathbf{t}), \theta_i, \sigma_i^2)\right\} p(\mu_0(\mathbf{t}) \mid \theta_0) \\ &= \prod_{i=1}^{M} \left\{\mathcal{N}(\mathbf{y}_i; \mu_0(\mathbf{t}), \boldsymbol{\Psi}_{\theta_i,\sigma_i^2})\right\} \mathcal{N}(\mu_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}_{\theta_0}^{\mathbf{t}}). \end{aligned}$$

The expectation is taken over $p(\mu_0(\mathbf{t}) \mid \{\mathbf{y}_i\}_i)$ though we write it $\mathbb{E}$ for simplicity. We have:

$$\begin{aligned} f(\Theta) &= \mathbb{E}\left[\log p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta)\right] \\ &= -\frac{1}{2}\mathbb{E}\Bigg[(\mu_0(\mathbf{t}) - m_0(\mathbf{t}))^{\intercal} {\mathbf{K}_{\theta_0}^{\mathbf{t}}}^{-1} (\mu_0(\mathbf{t}) - m_0(\mathbf{t})) - \log\left|{\mathbf{K}_{\theta_0}^{\mathbf{t}}}^{-1}\right| \\ &\quad + \sum_{i=1}^{M} (\mathbf{y}_i - \mu_0(\mathbf{t}_i))^{\intercal} {\boldsymbol{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1} (\mathbf{y}_i - \mu_0(\mathbf{t}_i)) - \log\left|{\boldsymbol{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1}\right|\Bigg] + C_1. \end{aligned}$$

Lemma 1 *Let $X \in \mathbb{R}^N$ be a random Gaussian vector $X \sim \mathcal{N}(m, \mathbf{K})$, $b \in \mathbb{R}^N$, and $\mathbf{S}$, a $N \times N$ covariance matrix. Then*:

$$\begin{aligned} E &= \mathbb{E}_X\left[(X - b)^{\intercal} \mathbf{S}^{-1} (X - b)\right] \\ &= (m - b)^{\intercal} \mathbf{S}^{-1} (m - b) + \mathrm{Tr}(\mathbf{K}\mathbf{S}^{-1}). \end{aligned}$$

Proof (Lemma 1)

$$\begin{aligned} E &= \mathbb{E}_X\left[\mathrm{Tr}(\mathbf{S}^{-1}(X - b)(X - b)^{\intercal})\right] \\ &= \mathrm{Tr}(\mathbf{S}^{-1}\mathbb{V}_X(X - b)) + \mathrm{Tr}(\mathbf{S}^{-1}(m - b)(m - b)^{\intercal}) \\ &= (m - b)^{\intercal}\mathbf{S}^{-1}(m - b) + \mathrm{Tr}\left(\mathbf{K}\mathbf{S}^{-1}\right). \end{aligned}$$

□

As we note that X and b play symmetrical roles in the calculus of the conditional expectation, we can apply the lemma regardless of the position of μ_0 in the $M + 1$ equalities involved. Applying Lemma 1 to our previous expression of $f(\Theta)$, we obtain:

$$\begin{aligned} f(\Theta) = -\frac{1}{2}\Bigg[& (\widehat{m}_0(\mathbf{t}) - m_0(\mathbf{t}))^{\intercal} {\mathbf{K}_{\theta_0}^{\mathbf{t}}}^{-1} (\widehat{m}_0(\mathbf{t}) - m_0(\mathbf{t})) \\ & + \sum_{i=1}^{M} (\mathbf{y}_i - \widehat{m}_0(\mathbf{t}_i))^{\intercal} {\boldsymbol{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1} (\mathbf{y}_i - \widehat{m}_0(\mathbf{t}_i)) \\ & + \mathrm{Tr}\left(\widehat{\mathbf{K}}^{\mathbf{t}} {\mathbf{K}_{\theta_0}^{\mathbf{t}}}^{-1}\right) + \sum_{i=1}^{M} \mathrm{Tr}\left(\widehat{\mathbf{K}}^{\mathbf{t}_i} {\boldsymbol{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1}\right) \\ & - \log\left|{\mathbf{K}_{\theta_0}^{\mathbf{t}}}^{-1}\right| - \sum_{i=1}^{M} \log\left|{\boldsymbol{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}}^{-1}\right| + C_1\Bigg]. \end{aligned}$$

We recall that, at the M step, $\widehat{m}_0(\mathbf{t})$ is a known constant, computed at the previous E step. Thus, we identify here the characteristic expression of several Gaussian log-likelihoods and associated correction trace terms. Moreover, each set of hyper-parameters is merely involved in independent terms of the whole function to maximise. Hence, the global maximisation problem can be separated into several maximisations of sub-functions according to the hyper-parameters getting optimised. Regardless to additional assumptions, the hyper-parameters θ_0, controlling the covariance matrix of the mean process, appears in a function which is exactly a Gaussian log-likelihood, $\log \mathcal{N}\left(\widehat{m}_0(\mathbf{t}), m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right)$, added to a corresponding trace term, $-\frac{1}{2}\mathrm{Tr}\left(\widehat{\mathbf{K}}^{\mathbf{t}}{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\right)$. This function can be maximised independently from the other parameters, giving the first part of the results in Propositions 2 and 3.

Although the idea is analogous for the remaining hyper-parameters, we have to discriminate here regarding the assumption on the model. If each individual is supposed to have its own set $\left\{\theta_i, \sigma_i\right\}$, which thus can be optimised independently from the observations and hyper-parameters of other individuals, we identify a sum of M Gaussian log-likelihoods, $\log \mathcal{N}\left(\mathbf{y}_i, \widehat{m}_0(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)$, and the corresponding trace terms, $-\frac{1}{2}\mathrm{Tr}(\widehat{\mathbf{K}}^{\mathbf{t}}{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1})$. This property results in M independent maximisation problems on the corresponding functions, proving Proposition 2. Conversely, if we assume that all individuals in the model shares their hyper-parameters (i.e. $\left\{\theta, \sigma^2\right\} = \left\{\theta_i, \sigma_i^2\right\}, \forall i \in \mathcal{I}$), we can no longer divide the problem into M sub-maximisations, and the whole sum on all individual should be optimised thanks to observations from all individuals. This case corresponds to the second part of Proposition 3. □

Acknowledgements The authors warmly thank Andy Marc, Olivier Dupas, Richard Martinez and the French Swimming Federation for providing data and helping in the analysis of the results. 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.

Funding The authors received no financial support for the research, authorship, and/or publication of this article.

Data availability The synthetic data and table of results are available at https://github.com/ArthurLeroy/MAGMA/tree/master/Simulations.

Code availability The R code associated with the present work is available at https://github.com/ArthurLeroy/MAGMA. The current version of the R package implementing an extended version of Magma is available at https://github.com/ArthurLeroy/MagmaClustR.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Alaa, A. M., & van der Schaar, M. (2017). Bayesian inference of individualized treatment effects using multi-task Gaussian processes. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.) *Advances in neural information processing systems 30*, Curran Associates, Inc., pp. 3424–3432.

Álvarez, M. A., & Lawrence, N. D. (2011). Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research, 12*(41), 1459–1500.

Álvarez, M. A., Rosasco, L., & Lawrence, N.D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning, 4*(3), 195–266. https://doi.org/10.1561/2200000036

Biernacki, C., Celeux, G., & Govaert, G. (2003). 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. https://doi.org/10.1016/S0167-9473(02)00163-9

Bishop, C. M. (2006). *Pattern recognition and machine learning, information science and statistics*. Springer.

Bonilla, E. V., Chai, K. M., & Williams, C. (2008). Multi-task Gaussian process prediction. In Platt, J. C., Koller, D., Singer, Y., Roweis, S. T. (Eds.) *Advances in neural information processing systems 20*, Curran Associates, Inc., pp. 153–160.

Caruana, R. (1997). Multitask learning. *Machine Learning, 28*(1), 41–75. https://doi.org/10.1023/A:1007379606734

Casella, G. (1985). An introduction to empirical Bayes data analysis. *The American Statistician, 39*(2), 83–87. https://doi.org/10.2307/2682801

Clingerman, C., & Eaton, E. (2017). Lifelong learning with Gaussian processes. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (Eds) *Machine learning and knowledge discovery in databases* (Vol. 10535, pp 690–704). Springer. https://doi.org/10.1007/978-3-319-71246-8_42

Crainiceanu, C. M., & Goldsmith, A. J. (2010). Bayesian functional data analysis using WinBUGS. *Journal of Statistical Software*, *32*(11).

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

Duvenaud, D. (2014). *Automatic model construction with Gaussian processes*. Thesis, University of Cambridge, https://doi.org/10.17863/CAM.14087

Ferraty, F., & Vieu, P. (2006). *Nonparametric functional data analysis: Theory and practice*. Springer.

Hayashi, K., Takenouchi, T., Tomioka, R., & Kashima, H. (2012). Self-measuring similarity for multi-task Gaussian process. *Transactions of the Japanese Society for Artificial Intelligence, 27*(3), 103–110. https://doi.org/10.1527/tjsai.27.103

McLachlan, G. J., & Krishnan, T. (2007). *The EM algorithm and extensions*. Wiley.

Morales, J. L., & Nocedal, J. (2011). 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. https://doi.org/10.1145/2049662.2049669

Moreno-Muñoz, P., Artés-Rodríguez, A., & Álvarez, M. A. (2019). Continual multi-task Gaussian processes. arXiv:1911.00002 [cs, stat] arXiv:1911.00002

Nguyen, T. V., & Bonilla, E. V. (2014). Collaborative multi-output Gaussian processes. In *Proceedings of the thirtieth conference on uncertainty in artificial intelligence*, AUAI Press, UAI'14, pp. 643–652

Nocedal, J. (1980). Updating quasi-Newton matrices with limited storage. *Mathematics of Computation, 35*(151), 773–782. https://doi.org/10.1090/S0025-5718-1980-0572855-7

Quiñonero-Candela, J., Rasmussen, C. E., & Williams, C. K. I. (2007). *Approximation methods for Gaussian process regression*. MIT Press.

Rakitsch, B., Lippert, C., Borgwardt, K., & Stegle, O. (2013). It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In *Advances in neural information processing systems 26*, Curran Associates, Inc., pp. 1466–1474

Ramsay, J. O., & Silverman, B. W. (2005). *Functional data analysis*. Springer.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning, adaptive computation and machine learning*. MIT Press.

Rice, J. A., & Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society Series B (Methodological), 53*(1), 233–243.

Schwaighofer, A., Tresp, V., & Yu, K. (2004). Learning Gaussian process kernels via hierarchical bayes. *Advances in Neural Information Processing Systems, 17*, 8.

Shi, J. Q., & Cheng, Y. (2014). Gaussian process function data analysis R package 'GPFDA'. https://cran.r-project.org/web/packages/GPFDA/GPFDA.pdf

Shi, J. Q., & Choi, T. (2011). *Gaussian process regression analysis for functional data*. CRC Press.

Shi, J., Murray-Smith, R., & Titterington, D. (2005). Hierarchical Gaussian process mixtures for regression. *Statistics and Computing, 15*(1), 31–41. https://doi.org/10.1007/s11222-005-4787-7

Shi, J. Q., Wang, B., Murray-Smith, R., & Titterington, D. M. (2007). Gaussian process functional regression modeling for batch data. *Biometrics, 63*(3), 714–723. https://doi.org/10.1111/j.1541-0420.2007.00758.x

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems* (Vol 18), MIT Press

Swersky, K., Snoek, J., & Adams, R. P. (2013). Multi-task Bayesian optimization. *Advances in Neural Information Processing Systems, 26*, 2004–2012.

Thompson, W. K., & Rosen, O. (2008). A Bayesian model for sparse functional data. *Biometrics, 64*(1), 54–63. https://doi.org/10.1111/j.1541-0420.2007.00829.x

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the twelth international conference on artificial intelligence and statistics, PMLR*, pp. 567–574.

Ueda, N., & Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural Networks, 11*(2), 271–282. https://doi.org/10.1016/S0893-6080(97)00133-0

Williams, C., Klanke, S., Vijayakumar, S., & Chai, K. M. (2009). Multi-task Gaussian process learning of robot inverse dynamics. *Advances in Neural Information Processing Systems, 21*, 265–272.

Yang, J., Zhu, H., Choi, T., & Cox, D. D. (2016). Smoothing and mean-covariance estimation of functional data with a Bayesian hierarchical model. *Bayesian Analysis, 11*(3), 649–670. https://doi.org/10.1214/15-BA967

Yang, J., Cox, D. D., Lee, J. S., Ren, P., & Choi, T. (2017). Efficient Bayesian hierarchical functional data analysis with basis function approximations using Gaussian-Wishart processes. *Biometrics, 73*(4), 1082–1091. https://doi.org/10.1111/biom.12705

Yu, K., Tresp, V., & Schwaighofer, A. (2005). Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on machine learning*, ACM, ICML '05, pp. 1012–1019. https://doi.org/10.1145/1102351.1102479

Zhu, J., & Sun, S. (2014). Multi-task sparse Gaussian processes with improved multi-task sparsity regularization. In *Pattern recognition*, Springer, pp. 54–62, https://doi.org/10.1007/978-3-662-45646-0_6

Authors and Affiliations

Arthur Leroy[1] · **Pierre Latouche**[2] · **Benjamin Guedj**[3,4] · **Servane Gey**[2]

Pierre Latouche
pierre.latouche@math.cnrs.fr
http://helios.mi.parisdescartes.fr/~platouch/

Benjamin Guedj
benjamin.guedj@inria.fr
https://bguedj.github.io

Servane Gey
servane.gey@parisdescartes.fr
http://helios.mi.parisdescartes.fr/~gey/

[1] Department of Computer Science, The University of Sheffield, Sheffield, UK

[2] Université de Paris, CNRS, MAP5 UMR 8145, 75006 Paris, France

[3] Inria (Lille - Nord Europe Research Centre), Lille, France

[4] University College London (Centre for Artificial Intelligence, Department of Computer Science), London, UK