Multi-Mean Gaussian Processes: A novel probabilistic framework for multi-correlated longitudinal data

Arthur Leroy *
Department of Computer Science
The University of Manchester
arthur.leroy.pro@gmail.com

Mauricio A Alvarez Lopez
Department of Computer Science
The University of Manchester
mauricio.alvarezlopez@manchester.ac.uk

1 Introduction

Modelling and forecasting time series, even with a probabilistic flavour, is a common and well-handled problem nowadays. In particular, Gaussian processes are by essence tailored to represent continuous phenomena, and several approaches have been proposed during the past decades to handle multiple tasks in one model. From the seminal work of Yu et al. [2005], or the definition of coregionalisation kernels [Goovaerts and Goovaerts, 1997, Bonilla et al., 2008], to the unifying view proposed in the subsequent review Álvarez et al. [2012], many approaches have been explored [Fortuin et al., 2019, van der Wilk et al., 2020], with the constraint that the number of tasks remain reasonable (the complexity generally scales cubically with the number of tasks). However, suppose now that one is collecting data from hundreds of individuals, each of them gathering thousands of gene-related measurements, all evolving continuously over time. Such a context, frequently arising in biological or medical studies, quickly leads to highly correlated datasets where dependencies come from different sources (temporal trend, gene or individual similarities for instance). Explicit modelling of overly large covariance matrices accounting for these underlying correlations is generally unreachable due to theoretical and computational limitations. Therefore, practitioners often need to restrict their analysis by working on subsets of data or making arguable assumptions (fixing time, studying genes or individuals independently, ...). Recently, a novel paradigm for defining multi-task Gaussian processes models has been proposed [Leroy et al., 2022] and tailored to handle multiple time series simultaneously. In this approach, a latent mean process common to all tasks is introduced, and knowledge is transferred through hyper-posterior computations and subsequent marginalisation. In this paper, we aim at offering an overview of this framework, and propose a more general formulation allowing us to handle multiple sources of correlations.

2 Sharing knowledge through a latent mean process

2.1 Modelling assumptions

Let us assume that longitudinal data are collected from M correlated tasks, over an arbitrary input space $\mathcal{T}$. To describe the generative model, we define a latent process common to all tasks, such that output for the i-th task can be expressed as:

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

where:

- $\mu_0(\cdot) \sim \mathcal{GP}(m_0(\cdot), K_{\theta_0}(\cdot, \cdot))$ is the common mean process,

*https://arthur-leroy.netlify.com/ - ORCID: 0000-0003-0806-8934

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

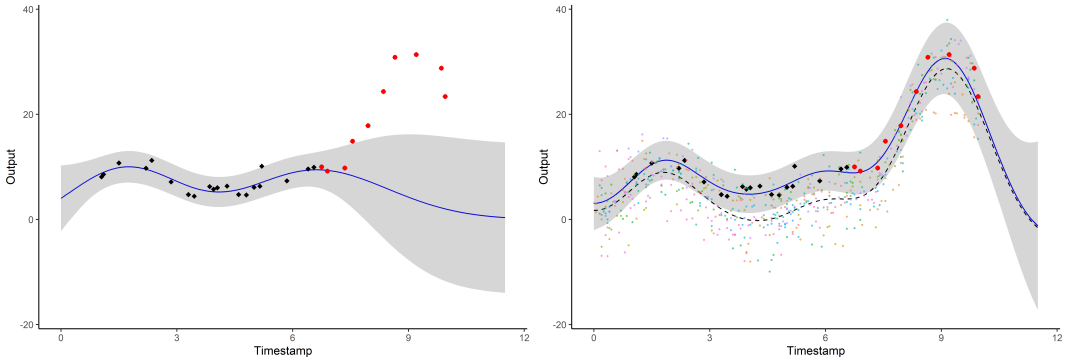

Figure 1: Prediction curves (blue) of a new task with associated 95% credible intervals (grey) for GP regression (left) and MAGMA (right). The dashed line represents the mean function $\hat{m}$ from the hyper-posterior. Observed points are in black and testing points in red. Colourful backward points are the observations from the training dataset, each colour corresponding to a different tasks. Figure from Leroy et al. [2022].

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

Under the hypothesis of independence between all pairwise elements, we remark that, conditionally to μ_0, all y_i are independent and the conditional likelihood remains Gaussian:

$$y_i(\cdot)|\mu_0(\cdot) \sim \mathcal{GP}(\mu_0(\cdot), \Sigma_{\gamma_i}(\cdot, \cdot) + \sigma_i^2 I).$$

2.2 Learning with an EM algorithm

In the model presented above, we both need to learn hyper-parameters of the kernels (as usual with GPs) but also compute the hyper-posterior distribution of the latent mean process μ_0. Although these quantities depend on one another, we can define an Expectation-Maximisation (EM) algorithm where the analytical derivation of the hyper-posterior of μ_0 is performed alternatively with hyper-parameters optimisation. During the E step, we can leverage the fact the hyper-posterior distribution remains Gaussian:

$$p(\boldsymbol{\mu}_0|\{\mathbf{y}_i\}_{i=1,\dots,M}) \propto p(\boldsymbol{\mu}_0) \prod_{i=1}^{M} p(\mathbf{y}_i|\boldsymbol{\mu}_0) = \mathcal{N}(\boldsymbol{\mu}_0; \hat{m}, \hat{K}), \tag{1}$$

with

- $\hat{K} = ({\mathbf{K}_{\theta_0}}^{-1} + \sum_{i=1}^{M} (\boldsymbol{\Sigma}_{\gamma_i} + \sigma_i^2 \mathbf{I})^{-1})^{-1}$,
- $\hat{m} = \hat{K}({\mathbf{K}_{\theta_0}}^{-1} \mathbf{m}_0 + \sum_{i=1}^{M} (\boldsymbol{\Sigma}_{\gamma_i} + \sigma_i^2 \mathbf{I})^{-1} \mathbf{y}_i)$.

Note that some of the above quantities are displayed in bold to indicate that we work on vectors for finite-dimensional evaluations of the underlying GPs.

2.3 Post-training marginalisation

Once the EM algorithm has converged after sufficient iterations of successive E and M steps, we are still unable to make predictions directly. The key idea of this method consists in marginalising the latent μ_0 *after* the training step. Assume that we aim at making predictions for an arbitrary task (the task may have been used for training or be newly observed), indexed by $*$. It is possible to derive a

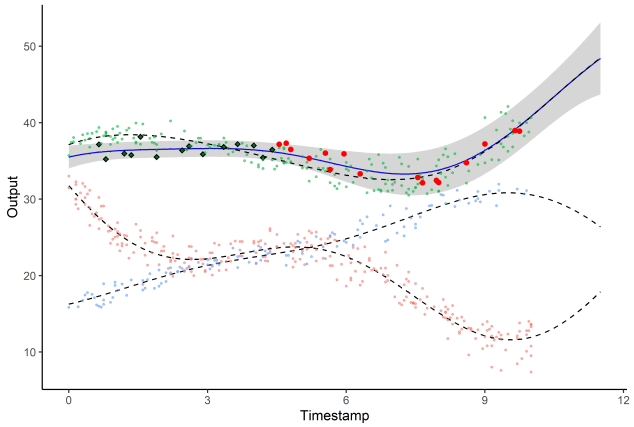

Figure 2: Prediction curves (blue) with associated 95% credible intervals (grey) for MAGMACLUST. The dashed lines represent the mean parameters from the mean processes estimates. Observed points are in black and testing points in red. Backward points are observations from the training tasks, coloured according to their most probable cluster. Figure from Leroy et al. [2020].

multi-task prior distribution for $\mathbf{y}_*$, the vector of interest:

$$\begin{aligned} p(\mathbf{y}_*|\{\mathbf{y}_i\}_{i=1,\dots,M}) &= \int p(\mathbf{y}_*, \boldsymbol{\mu}_0|\{\mathbf{y}_i\}_{i=1,\dots,M})\, \mathrm{d}\boldsymbol{\mu}_0 \\ &= \int p(\mathbf{y}_*|\boldsymbol{\mu}_0) p(\boldsymbol{\mu}_0|\{\mathbf{y}_i\}_{i=1,\dots,M})\, \mathrm{d}\boldsymbol{\mu}_0 \\ &= \mathcal{N}(\mathbf{y}_*;\ \hat{m}, \hat{K} + \boldsymbol{\Sigma}_{\gamma_*} + \sigma_*^2 I). \end{aligned}$$

One can notice that this expression involves the parameters $\hat{m}$ and $\hat{K}$, which have previously been learned thanks to the EM algorithm. Intuitively, defining this multi-task distribution allows us to implicitly transfer knowledge by defining an *already-informed* prior mean, built with data from multiple sources. To conclude the prediction step, one only needs to apply the classical formula for conditioning over Gaussian vectors, as usual in GP regression. This overall algorithm has been called MAGMA (standing for Multi-tAsk Gaussian processes with common MeAn). Let us display on Figure 1 an illustration of the predictive performance of MAGMA compared to standard GP regression. The method can take advantage of observations from other tasks to forecast a meaningful mean trend even far from data points. Additionally, the uncertainty quantification naturally takes into account observations both from the predicted task and the others. It is worth noticing that, by sharing information between tasks through a mean process instead of an explicit covariance structure, this method leads to a learning and forecasting procedure with linear complexity in the number of tasks. Moreover, the resulting predictions remain Gaussian and thus offer an elegant probabilistic framework to deal with correlated time series.

3 Simultaneous clustering of tasks

A valuable extension of the previous model has been proposed in Leroy et al. [2020] to perform a simultaneous clustering of the tasks during the learning step. More specifically, this is achieved by defining a mixture of Gaussian processes by assuming that if the i-th task belongs to the k-th cluster, then:

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

Although additional technical details are needed to derive learning and prediction formula, the intuition remains similar, and we display on Figure 2 the advantage of this extension (named MAGMACLUST). We can observe that the algorithm automatically allocates the similar tasks into clusters. Additionally, one mean process is now defined for each cluster, allowing the information transfer to be weighted depending on the similarities between tasks. In terms of implementation, the R package *MagmaClustR*, which has recently been released (https://github.com/ArthurLeroy/

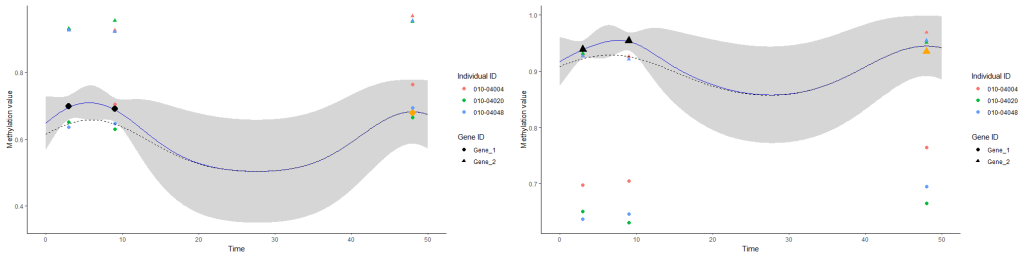

Figure 3: Prediction curves (blue) with associated 95% credible intervals (grey). The dashed lines represent the mean parameters from the multi-mean processes estimates. Observed points are in black and testing points in orange. Backward points are observations from the training tasks, where individuals are differentiated with colors and genes with shapes.

`MagmaClustR`), offers all the tools to perform learning, prediction and plotting of results for such models. Despite many appealing properties, this framework still presents the limitation to only account for one source of correlation (recalling our introducing example, we could handle multiple individuals or multiple genes but not both).

4 Multiple sources of correlation

In this work, we propose an extended framework in which as many sources of correlation as desired can be considered. As for the precedent formulation, we can express the generative model for two sources of correlation as:

$$y_{i,j}(t) = \mu_0(t) + f_i(t) + g_j(t) + \varepsilon_{i,j}(t),\ \forall t \in \mathcal{T},$$

where the additional process $g_j(\cdot) \sim \mathcal{GP}(0, \Sigma_{\gamma_j}(\cdot,\cdot))$ accounts for the specific behaviour of the j-th task of the second source of correlation. To illustrate with our previous example, we would say that our observed time series $y_{i,j}$ is composed of a common mean trend, summed with a perturbation specific to the i-th individual, and a perturbation specific to the j-th gene. Although the model is presented here with two sources of correlations for the sake of clarity, the formulation can be extended to an arbitrary number of additional terms. We also omitted the clustering aspect for simplicity, though it could also be considered with no particular difficulties.

The key idea in this context comes from the definition of multiple hyper-posterior distributions for μ_0 (the method is hence called *multi-mean GPs*) by selecting an adequate sub-sample of the training data set. Recalling the E step in MAGMA, we can now compute $M + N + 1$ distinct hyper-posterior distributions, defined as:

- $p(\boldsymbol{\mu}_0|\{\mathbf{y}_{i,j}\}_{i=1,\dots,M}),\ \forall j = 1,\dots,N,$
- $p(\boldsymbol{\mu}_0|\{\mathbf{y}_{i,j}\}^{j=1,\dots,N}),\ \forall i = 1,\dots,M,$
- $p(\boldsymbol{\mu}_0|\{\mathbf{y}_{i,j}\}_{i=1,\dots,M}^{j=1,\dots,N}).$

The training steps in this context remains tractable and the overall EM algorithm can be parallelised across the different hyper-posterior distributions to speed-up computations. Then, despite some technical details, the prediction step remains roughly analogous to what we previously detailed with MAGMA. However, *multi-mean GPs* leads to an adaptive predictive distribution, which uses the mean process that is specific to the considered correlation to improve forecasting abilities. An illustration of the multi-mean GPs model is proposed in Figure 3, where the prediction of methylation values over time is displayed, considering multiple individuals and genes. We can notice that, for each panel (left or right), a different mean process is used for prediction, depending on the gene that is considered. Thanks to the learning of the different multi-mean processes, the prediction can be adapted with the corresponding gene to provide remarkably accurate forecasts. Let us note that it remains possible to derive analytical predictions for a completely new individual or gene (although in this case, we would use the overall mean process, which might be less specific and thus lead to less accurate results).

References

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

E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-task Gaussian Process Prediction. *Advances in Neural Information Processing Systems 20*, pages 153–160, 2008.

V. Fortuin, H. Strathmann, and G. Rätsch. Meta-learning mean functions for gaussian processes. *arXiv preprint arXiv:1901.08098*, 2019.

P. Goovaerts and D. o. C. a. E. E. P. Goovaerts. *Geostatistics for Natural Resources Evaluation*. Oxford University Press, 1997. ISBN 978-0-19-511538-3.

A. Leroy, P. Latouche, B. Guedj, and S. Gey. Cluster-Specific Predictions with Multi-Task Gaussian Processes. *arXiv:2011.07866 [cs, stat]*, Nov. 2020.

A. Leroy, P. Latouche, B. Guedj, and S. Gey. MAGMA: Inference and prediction using multi-task Gaussian processes with common mean. *Machine Learning*, May 2022. ISSN 1573-0565. doi: 10.1007/s10994-022-06172-1.

M. van der Wilk, V. Dutordoir, S. John, A. Artemev, V. Adam, and J. Hensman. A framework for interdomain and multioutput gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 1012–1019, New York, NY, USA, Aug. 2005. Association for Computing Machinery. ISBN 978-1-59593-180-1. doi: 10.1145/1102351.1102479.