Probabilistic Graphical Models and Deep Generative Models
MAGMA: inference and prediction using multi-task Gaussian processes with common mean

Ben Kabongo ben.kabongo_buzangu@ens-paris-saclay.fr
Hugo Queniat hugo.queniat@telecom-paris.fr
Paul Castéras paul.casteras@student-cs.fr
Master MVA, ENS Paris-Saclay

December 2023

1 Introduction

1.1 Presentation of the article

We worked on the article **MAGMA: inference and prediction using multi-task Gaussian processes with common mean** written by Arthur Leroy, Pierre Latouche, Benjamin Guedj, and Servane Gey [1].

The model proposed in this article is a multi-task Gaussian process, applied to time series forecasting, where processes don't share a common covariance matrix and zero mean, as in most previous work, but share a common mean. The model is trained with the EM algorithm to calculate its parameters.

Our implementation is available here : https://github.com/BenKabongo25/mva-pgm-project.

1.2 Notations and Hypotheses

In the following, we will use the term **individual** to refer to a single task or process. In the context of multi-task Gaussian processes, **M** is the number of individuals.

We have a set of timestamps $\mathbf{t}_i = \{t_i^1, \dots, t_i^{Ni}\}$ and associated outputs $\mathbf{y}_i(\mathbf{t}_i) = \{y_i(t_i^1), \dots, y_i(t_i^{Ni})\}$, for each individual i. We set $\mathbf{t} = \cup \mathbf{t}_i$. We call a configuration in which all individuals share the same timestamps a **common grid**; otherwise, it's called an **uncommon grid** configuration.

It is assumed that $\mathbf{y}_i$ can be decomposed into the following terms:

$$\mathbf{y}_i(t) = \mu_0(t) + f_i(t) + \varepsilon_i(t)$$

with :

- $\mu_0(\cdot) \sim \mathcal{GP}(m_0(\cdot), k_{\theta_0}(\cdot,\cdot))$ a common mean with m_0 a prior mean function and k_{θ_0} a kernel function of hyper-parameters θ_0.
- $f_i(\cdot) \sim \mathcal{GP}(0, c_{\theta_i}(\cdot,\cdot))$ a centered gaussian process that is different for each individual. c_{θ_i} is a kernel function of hyper-parameters θ_i.
- $\varepsilon_i(\cdot) \sim \mathcal{N}(0, \sigma_i^2 \mathbf{I})$ an error term.

To simplify notations let $\Psi_{\theta_i, \sigma_i^2}(\cdot,\cdot) = c_{\theta_i}(\cdot,\cdot) + \sigma_i^2 \mathbf{I}$.

We call a configuration in which all individuals share the same parameters θ_i and σ_i^2 **common HP**; otherwise, it's called a **different HP** configuration.

We will explain the model with the exponential quadratic kernel : $k(x, x') = v^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$, but the model can easily be modified to use other kernels.

2 MAGMA model

The MAGMA hyper-parameters to be optimized are : $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$. These hyper-parameters can then be optimized using an EM algorithm. The E step consists in determining the distribution of $p(\mu_0 \mid \{\mathbf{y}_i\}_i, \hat{\Theta})$ with the current hyper-parameters. The M step consists in optimizing the hyper-parameters Θ with a constant distribution $p(\mu_0 \mid \{y_i\}_i, \hat{\Theta})$. Once the model's hyper-parameters have been learned, predictions can be made for new individuals.

2.1 EM algorithm for learning hyper-parameters

2.1.1 E step

The hyper-posterior distribution of μ_0 remains gaussian:

$$p(\mu_0 \mid \{\mathbf{y}_i\}_i, \hat{\Theta}) = \mathcal{N}(\mu_0(\mathbf{t}); m_0(\mathbf{t}), \hat{\mathbf{K}}^{\mathbf{t}})$$

with:

- $\hat{\mathbf{K}}^{\mathbf{t}} = \left({\mathbf{K}_{\hat{\theta}_0}^{\mathbf{t}}}^{-1} + \sum_{i=1}^{M} {\Psi_{\hat{\theta}_i, \hat{\sigma}_i^2}^{\mathbf{t}}}^{-1} \right)^{-1}$
- $\hat{m}_0(\mathbf{t}) = \hat{\mathbf{K}}^{\mathbf{t}} \left(\hat{\mathbf{K}}^{\mathbf{t}-1} m_0(\mathbf{t}) + \sum_{i=1}^{M} {\Psi_{\hat{\theta}_i, \hat{\sigma}_i^2}^{\mathbf{t}}}^{-1} \mathbf{y}_i \right)$

2.1.2 M step

The problem to be solved in step M is as follows :

$$\hat{\Theta} = \arg\max_{\Theta} \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i} \left[p\left(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta \right) \right]$$

This is equivalent to solving the following sub-problems:

$$\hat{\theta}_0 = \arg\max_{\theta_0} \mathcal{L}^{\mathbf{t}} \left(\hat{m}_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}_{\hat{\theta}_0}^{\mathbf{t}} \right) = \arg\max_{\theta_0} g(\theta_0) \tag{1}$$

and M other optimization problems in the case of a different HP configuration:

$$\forall i, (\hat{\theta}_i, \hat{\sigma}_i^2) = \arg\max_{\theta_i, \sigma_i^2} \mathcal{L}^{\mathbf{t}_i} \left(\mathbf{y}_i; \hat{m}_0(\mathbf{t}), \Psi_{\theta_i, \sigma_i^2}^{\mathbf{t}_i} \right) = \arg\max_{\theta_i, \sigma_i^2} h_i(\theta_i, \sigma_i^2) \tag{2}$$

or a single optimization problem in the case of a common HP configuration:

$$(\hat{\theta}, \hat{\sigma}^2) = \arg\max_{\theta, \sigma^2} \sum_{i=1}^{M} \mathcal{L}^{\mathbf{t}_i} \left(\mathbf{y}_i; \hat{m}_0(\mathbf{t}), \Psi_{\theta, \sigma^2}^{\mathbf{t}_i} \right) = \arg\max_{\theta, \sigma^2} h(\theta, \sigma^2) \tag{3}$$

where $\mathcal{L}^{\mathbf{t}}(\mathbf{x}; \mathbf{m}, \mathbf{S}) = \log \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{S}) - \frac{1}{2} \mathrm{Tr}(\hat{\mathbf{K}}^{\mathbf{t}} \mathbf{S}^{-1})$.

To solve these problems, we need to use gradient-based methods. Hence, we computed the gradients for each of this objective functions, with the computations being recalled in the appendix A.1.2.

2.2 Prediction

The previous section focused on the training of the model to learn the hyper-parameters that fit best the input data, we will now discuss the prediction of a new individual $*$ from which observed the data at time points $\mathbf{t}_* : \{\mathbf{t}_*, y_*(\mathbf{t}_*)\}$. The goal is to predict the evolution of the individual $*$ through our MAGMA model at timestamps $\mathbf{t}^p$. A distinction is made between **Type I** and **Type II** prediction. Type I corresponds to a case of prediction for which some data have been observed, and type II to a case where no data have been observed. Type II can be considered a special case of Type I.

In order to make predictions with MAGMA , it is necessary to compute the posterior inference on the mean process $\mu_0(.)$ at the new timestamps that are now being examined, $\mathbf{t}_*$ the observed timestamps for the new individual and $\mathbf{t}^p$ the time points of prediction. Hence, we consider the pooled set of timestamps : $\mathbf{t}_*^p = \mathbf{t}_* \cup \mathbf{t}^p$. The distribution of $\mu_0(.)$ remains Gaussian:

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

with:

- $\widehat{\mathbf{K}}_*^p = \left(\tilde{\mathbf{K}}^{-1} + \sum_{i=1}^M \tilde{\Psi}_i^{-1}\right)^{-1}$
- $\widehat{m}_0\left(\mathbf{t}_*^p\right) = \widehat{\mathbf{K}}_*^p\left(\tilde{\mathbf{K}}^{-1} m_0\left(\mathbf{t}_*^p\right) + \sum_{i=1}^M \tilde{\Psi}_i^{-1}\tilde{\mathbf{y}}_i\right)$
- $\tilde{\mathbf{K}} = k_{\hat{\theta}_0}\left(\mathbf{t}_*^p, \mathbf{t}_*^p\right)$
- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t\in t_i]} \times y_i(t)\right)_{t\in\mathbf{t}_*^p}$
- $\tilde{\Psi}_i = \left[\mathbb{1}_{[t,t'\in t_i]} \times \psi_{\widehat{\theta}_i,\hat{\sigma}_i^2}(t,t')\right]_{t,t'\in\mathbf{t}_*^p}$

Given a set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of $\mathbf{y}_*$ is given by:

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

where:

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

Hence, we get the optimization problem to learn the new parameters from the data obtained over the new individual $\{\mathbf{t}_*, y_*(\mathbf{t}_*)\}$ by using the gaussian vector property :

$$\widehat{\Theta}_* = \arg\max_{\Theta_*} \mathcal{N}\left(y_*(\mathbf{t}_*); \widehat{m}_0(\mathbf{t}_*), \mathbf{\Gamma}_{**}^{\Theta_*}\right).$$

This step is not necessary if we are in a common HP configuration since the new individual's configuration will carry the common hyperparameters learned for the previous individuals.

Finally, this allows us to compute the posterior distribution for the prediction of $(y(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$:

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

with

- $\widehat{\mu}_0^p = \widehat{m}_0(\mathbf{t}^p) + \mathbf{\Gamma}_{p*}\mathbf{\Gamma}_{**}^{-1}\left(y_*(\mathbf{t}_*) - \widehat{m}_0(\mathbf{t}_*)\right)$
- $\widehat{\mathbf{\Gamma}}^p = \mathbf{\Gamma}_{pp} - \mathbf{\Gamma}_{p*}\mathbf{\Gamma}_{**}^{-1}\mathbf{\Gamma}_{*p}$

3 Experiments and Results

We tested our algorithm on both synthetic and real datasets. We tested all configurations for the synthetic dataset (common or uncommon grid, common HP or different HP). For the real dataset, we chose a specific configuration based on the dataset properties.

3.1 Experiments on synthetic data

3.1.1 Configuration comparison

In order to compare the different configurations of the MAGMA model with each other, we generate synthetic data for the common HP and different HP hyper-parameter configurations, and for each, we study the common grid and uncommon grid cases. We set a common μ_0 average for all four configurations. The training data are generated according to the configurations. For each configuration, we have a model MAGMA that is trained, for the same maximum number of iterations. For all four configurations, prediction is then studied on the same data. We compare configurations in the sense of loss MSE, and report the results in Table 1.

		Estimation of μ_0	Prediction
Common HP	Common grid	65.867	418.062
	Uncommon grid	67.789	132.734
Different HP	Common grid	38.557	63.293
	Uncommon grid	149.251	57.474

Table 1: Loss MSE of μ_0 mean estimation by MAGMA and loss MSE of prediction on synthetic data, for the different possible model configurations.

The predictions for the different HP case are better than the common HP case, because the data used to evaluate the prediction of the models all have different hyper-parameters. But we can't say which configuration best estimates the mean μ_0; indeed, we soon realize that this is intrinsically linked to the nature and type of data.

3.1.2 Type I and Type II prediction

We followed the scheme described in section 6 of the article to generate the synthetic data. The figures 1 and 2 are a prediction example for prediction of Type I and II in the case uncommon grid and different HP. In this example, we obtain better results for Type I prediction. This confirms that Type I is more informative than Type II.

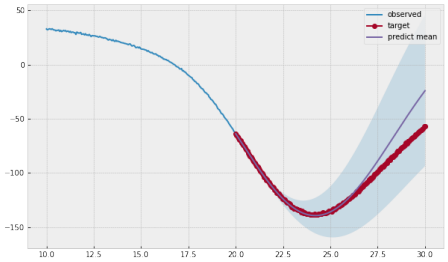


Figure 1: A prediction of type I

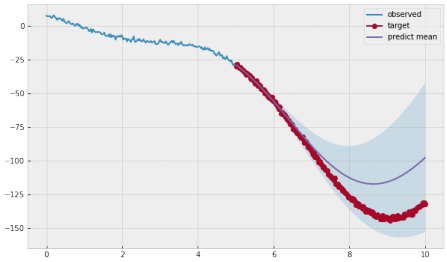


Figure 2: A prediction of type II

3.2 Experiments on real datasets

3.2.1 Comparison between MAGMA and GP

			MAGMA	GP
ECG5000	M = 5	Common HP	0.366 (0.033)	0.400 (2.331×10^{-15})
		Different HP	0.356 (0.019)	0.400 (2.331×10^{-15})
	M = 10	Common HP	0.366 (0.020)	0.376 (1.276×10^{-15})
		Different HP	0.360 (0.053)	0.376 (1.276×10^{-15})
	M = 20	Common HP	0.361 (0.016)	0.374 (1.110×10^{-16})
		Different HP	0.359 (0.064)	0.374 (1.110×10^{-16})
Traffic	M = 5	Common HP	6.005×10^{-3} (2.768×10^{-4})	6.009×10^{-3} (3.642×10^{-17})
		Different HP	5.988×10^{-3} (1.775×10^{-4})	6.009×10^{-3} (3.642×10^{-17})
	M = 10	Common HP	5.995×10^{-3} (2.152×10^{-4})	6.021×10^{-3} (3.729×10^{-17})
		Different HP	5.962×10^{-3} (1.574×10^{-4})	6.021×10^{-3} (3.729×10^{-17})
	M = 20	Common HP	6.000×10^{-3} (1.647×10^{-4})	6.009×10^{-3} (1.648×10^{-17})
		Different HP	3.228×10^{-1} (6.837)	6.009×10^{-3} (1.648×10^{-17})

Table 2: MSE loss of MAGMA and GP models on ECG5000 and Traffic datasets. **The mean of individuals for prediction is given by the dataset mean**. The mean MSE value and standard deviation are given in brackets. MAGMA *'s MSE is always lower than that of GPs.*

			MAGMA	GP
ECG5000	M = 5	Common HP	1.154 (0.441)	1.195 (0.440)
		Different HP	1.147 (0.439)	1.195 (0.440)
	M = 10	Common HP	1.156 (0.441)	1.171 (0.440)
		Different HP	1.149 (0.441)	1.171 (0.440)
	M = 20	Common HP	1.150 (0.440)	1.169 (0.441)
		Different HP	1.148 (0.442)	1.169 (0.441)
Traffic	M = 5	Common HP	8.687×10^{-3} (5.005×10^{-3})	8.721×10^{-3} (4.962×10^{-3})
		Different HP	8.664×10^{-3} (4.997×10^{-3})	8.721×10^{-3} (4.962×10^{-3})
	M = 10	Common HP	8.678×10^{-3} (4.993×10^{-3})	8.733×10^{-3} (4.969×10^{-3})
		Different HP	8.641×10^{-3} (4.990×10^{-3})	8.733×10^{-3} (4.969×10^{-3})
	M = 20	Common HP	8.682×10^{-3} (4.988×10^{-3})	8.722×10^{-3} (4.973×10^{-3})
		Different HP	8.699×10^{-3} (5.017×10^{-3})	8.722×10^{-3} (4.973×10^{-3})

Table 3: MSE loss of MAGMA and GP models on ECG5000 and Traffic datasets. **Each individual for which the prediction is made is its own mean.** The mean MSE value and standard deviation are given in brackets. MAGMA *'s MSE is always lower than that of GPs.*

In a first set of experiments, we compare the MAGMA and Gaussian Process models, that we have also implemented, in the sense of the MSE loss, by varying the parameter configuration (common/different HP) and the number of individuals. To avoid giving an advantage to MAGMA , which is trained on several individuals, we will compare each time a MAGMA model with M individuals to M Gaussian processes, each trained on an individual. The models are initialized with the same parameters. The models are evaluated with the MSE loss between predicted and real mean. In some cases, we consider an individual's output to be its mean value. The results of the Gaussian Processes are then averaged and compared with those of the MAGMA model.

We carry out our experiments on real data, using the ECG5000[1] and Traffic[2] datasets, which are time-series forecast datasets. For both datasets, the models are trained on 60% of each individual, and predictions are made and evaluated on the remaining 40%.

After training the models on a number of individuals from the dataset, we use the models to make predictions on the entire dataset. In the first instance, we consider the real mean to be the mean over the whole dataset (see Table 2). In the second case, each individual is considered to be its mean (see Table 3). In both cases, MAGMA has a lower average MSE than GP models. We also note that the difference between the two models is not very large. But since MAGMA performs better, we can conclude that using a common mean in GP multi-tasking is relevant.

4 Discussion

To begin with, we would like to point out the fact that our implementation is not an exact transcription of the model MAGMA theoretically described in [1]. Indeed, to deal with potential issues concerning numerical stability we've had to implement approximates of the computations we presented.
Mainly, we added to most covariance matrices a small matrix $\varepsilon \mathbf{I}$ with $\varepsilon > 0$, to ensure that all eigenvalues are positive and the matrices remain non-singular.

Secondly, by looking closely to the experiments we did in the previous section, we can discuss one of the main differences between the models we compared. Indeed, the model we chose to compare MAGMA to, the traditional Gaussian Process reduction with a reduction for each individual, exhibits its most importance distinction in the study of the mean of the individuals. If the MSE loss for MAGMA proves to be lower than for GP in the mean study, we observe a massive gap in the standard deviations. There is literally no standard deviation (magnitude least or equal to 10^{-15}) for GP whereas we observe a really low but still existent std for the MAGMA (magnitude higher than 10^{-4}).
This observation is a direct consequence of the very essence of the MAGMA model. Indeed, it introduces a common Gaussian Process to model the mean instead of computing a fixed mean for each individual i such as the GP model does. As a result, GP captures really well the actual mean for each individual whereas MAGMA showcases a "higher" variance such that the common mean Gaussian Process remains relevant to any individual. This gap in variance shows directly on our box-plots, we observe outliers more frequently for the MAGMA model than for GP.

To finish with, we would like to discuss the effective performance of the MAGMA model. We have to acknowledge that the experiments we have performed on distinct datasets have shown an undeniable improvement in terms of accuracy against the usual Gaussian Process regression. However, we must balance this observation against the computational cost of the method. Indeed, the relative improvement observed was no larger than around 3% meanwhile computing times for the training of the model *at least* doubled - or even more - for MAGMA . Thus, when dealing with massive datasets - either numerous individuals or numerous timestamps -, one may wonder whether the improved accuracy provided by MAGMA is sufficiently worth the excess computational cost.

References

[1] Arthur Leroy, Pierre Latouche, Benjamin Guedj, and Servane Gey. Magma: inference and prediction using multi-task gaussian processes with common mean. *Machine Learning*, 111, 2022.

[1] https://timeseriesclassification.com/description.php?Dataset=ECG5000
[2] https://zenodo.org/records/4656132

A Proofs

A.1 Computing the gradients for the optimization problems

A.1.1 Matrix Calculus

In this first subsection, we recall several usual derivatives in matrix calculus, with $\mathbf{y} \in \mathbb{R}^n, \mathbf{X}, \mathbf{M} \in \mathbb{R}^{n\times n}$ with n being a positive integer :

$$\begin{aligned}\frac{\partial(\mathbf{y}^T\mathbf{X}\mathbf{y})}{\partial\mathbf{X}} &= \mathbf{y}^T\mathbf{y}\\ \frac{\partial \log\det(\mathbf{X})}{\partial\mathbf{X}} &= (\mathbf{X}^{-1})^T\\ \frac{\partial \operatorname{tr}(\mathbf{M}\mathbf{X})}{\partial\mathbf{X}} &= \mathbf{M}^T\\ \mathrm{d}\mathbf{X}^{-1} &= -\mathbf{X}^{-1}\,\mathrm{d}\mathbf{X}\,\mathbf{X}^{-1}\end{aligned}$$

A.1.2 Derivation of hyper-parameters

We will now compute the different partial derivatives necessary for the use of gradient-methods in solving the optimizations problems of the EM algorithm 2.1.2.

Our computations will rely heavily on the results recalled in A.1.1, while keeping in mind that we are here dealing with covariance kernels, meaning that all the matrices $\mathbf{K}$ and $\mathbf{\Psi}$ are symmetric.

We have three distinct optimization problems to solve, and we need to compute the associated partial derivatives for the gradient-descents.

- First, we concentrate on problem (1) and compute the derivative with respect to θ_0 of g:

$$g(\theta_0) = \log\mathcal{N}\left(\hat{m}_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right) - \frac{1}{2}\operatorname{tr}(\widehat{\mathbf{K}}^{\mathbf{t}}{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1})$$

  To do so, we will apply the chain rule and, thus, compute several intermediate partial derivatives.

$$\begin{aligned}\frac{\partial g}{\partial {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}} &= \frac{1}{2}\frac{\partial \log\det({\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1})}{\partial {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}} - \frac{1}{2}\frac{\partial\left((\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t}))^T{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}(\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t}))\right)}{\partial {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}} - \frac{1}{2}\frac{\partial \operatorname{tr}(\widehat{\mathbf{K}}^{\mathbf{t}}{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1})}{\partial {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}}\\ &= \frac{1}{2}\left(\mathbf{K}^{\mathbf{t}}_{\theta_0} - (\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t}))^T(\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t})) - \widehat{\mathbf{K}}^{\mathbf{t}}\right)\end{aligned}$$

  Then,

$$\begin{aligned}\frac{\partial g}{\partial \mathbf{K}^{\mathbf{t}}_{\theta_0}} &= -{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\frac{\partial g}{\partial k^{-1}_{\theta_0}}{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\\ &= \frac{1}{2}\left({\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}(\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t}))^T(\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t})){\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1} + {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\widehat{\mathbf{K}}^{\mathbf{t}}{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1} - {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\right)\\ &= \left(\frac{\partial g}{\partial(\mathbf{K}^{\mathbf{t}}_{\theta_0})_{t,t'}}\right)\end{aligned}$$

  Hence, by applying the chain rule we get :

$$\frac{\partial g}{\partial\theta_0} = \sum_t\sum_{t'}\frac{\partial g}{\partial(\mathbf{K}^{\mathbf{t}}_{\theta_0})_{t,t'}}\frac{\partial(\mathbf{K}^{\mathbf{t}}_{\theta_0})_{t,t'}}{\partial\theta_0}$$

- We now focus on (2) and compute the partial derivatives of h_i with respect to θ_i σ_i^2:

$$h_i(\theta_i, \sigma_i^2) = \log \mathcal{N}\left(\mathbf{y}_i; \hat{m}_0(\mathbf{t}), \Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right) - \frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{K}}^{\mathbf{t}}{\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right)$$

Once again, we compute intermediate partial derivatives much like the previous computation,

$$\begin{aligned}\frac{\partial h_i}{\partial \Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}} &= \frac{1}{2}\left({\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\hat{m}_0(\mathbf{t}) - y_i)^T(\hat{m}_0(\mathbf{t}) - y_i){\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1} + {\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\widehat{\mathbf{K}}^{\mathbf{t}}{\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1} - {\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right) \\ &= \left(\frac{\partial h_i}{\partial\left(\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)_{t,t'}}\right)\end{aligned}$$

We now can apply the chain rule and get :

  - With respect to $\theta_i = \{\theta_i^{(1)}, \cdots, \theta_i^{(P)}\}$:

$$\forall p \in \{1, \cdots, P\},\ \frac{\partial h_i}{\partial \theta_i^{(p)}} = \sum_t \sum_{t'} \frac{\partial h_i}{\partial\left(\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)_{t,t'}} \frac{\partial\left(\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)_{t,t'}}{\partial \theta_i^{(p)}}$$

  - With respect to σ_i^2 :

$$\frac{\partial h_i}{\partial \sigma_i^2} = \sum_t \sum_{t'} \frac{\partial h_i}{\partial\left(\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)_{t,t'}} \frac{\partial\left(\Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)_{t,t'}}{\partial \sigma_i^2}$$

- For the last problem (3), with $h(\theta, \sigma^2)$, we observe:

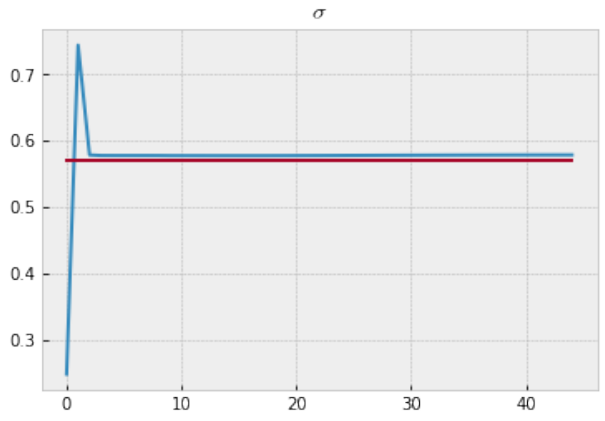
$$h(\theta, \sigma^2) = \sum_{i=1}^{M} h_i(\theta, \sigma^2)$$

Therefore, we can use the computations we did in the previous point for the M sub problems.

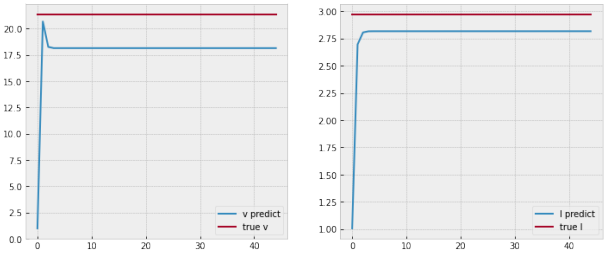
B Hyper-parameters convergence

B.1 Common HP

(a) Hyper-parameter σ



(b) Hyper-parameters θ



(c) Hyper-parameters θ_0

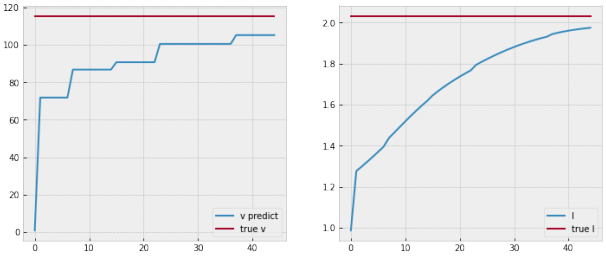
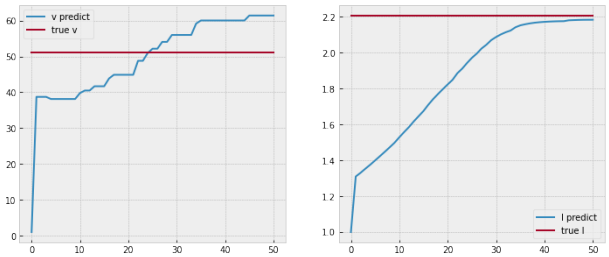


Figure 3: Example of the evolution of MAGMA model hyper-parameters (in blue) over real hyper-parameters (in red) in a *common HP* configuration.

B.2 Different HP

(a) Hyper-parameters θ_0



(b) Hyper-parameters θ_i

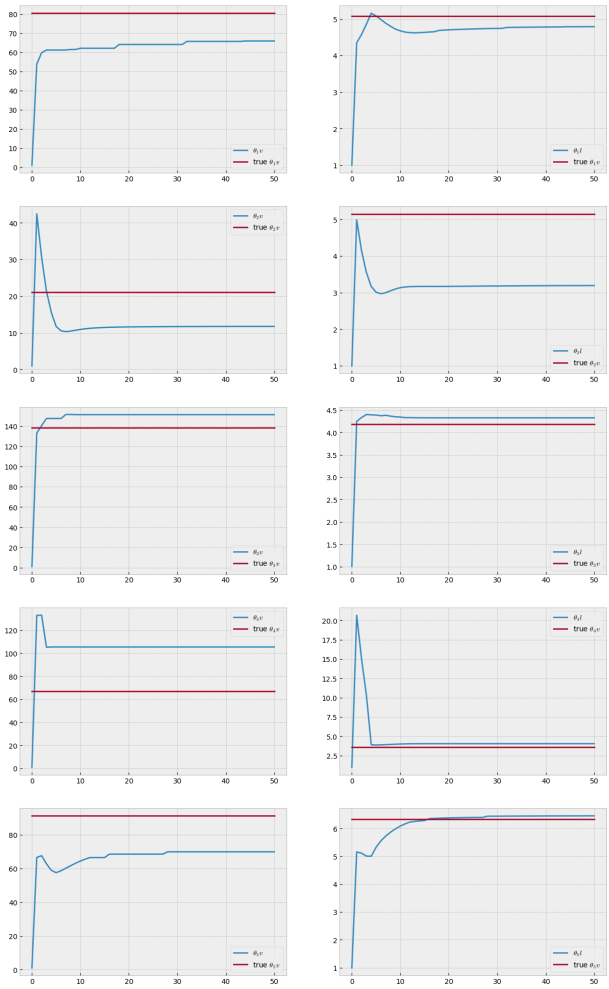


Figure 4: Example of the evolution of MAGMA model hyper-parameters (in blue) over real hyper-parameters (in red) in a *different HP* configuration.