PARALLEL COMPUTATION OF GAUSSIAN PROCESSES

R. Preuss, U. von Toussaint
Max-Planck-Institute for Plasma Physics
85748 Garching, Germany

April 20, 2017

Abstract

Within the Bayesian framework we utilize Gaussian processes for parametric studies of long running computer codes. Since the simulations are expensive it is necessary to exploit the computational budget in the best possible manner. Employing the sum over variances – being indicators for the quality of the fit – as the utility function we established an optimized and automated sequential parameter selection procedure. However, often it is also desirable to utilize the parallel running capabilities of present computer technology and abandon the sequential parameter selection for a faster overall turn-around time (wall-clock time). The paper proposes to achieve this by marginalizing over the expected outcomes at optimized test points in order to set up a pool of starting values for batch execution.

Keywords: Parametric studies, Gaussian process, parallelization, batch execution
PACS: 02.50.-r, 52.65.-y

1 Introduction

The prediction of optimal simulation input points is of particular interest for long running computer codes. As in the field of plasma-wall interactions of fusion plasmas the running time of the simulation codes is in the order of months, so the input parameter settings should better be chosen well. Considerable parallelization efforts have been spent to accelerate the code but the speed up is limited to a certain margin from which on it is futile to utilize more and more processor cores. So, even if more computer power is at hand, one would still have to wait for a sequential execution of one parameter setting after the other. The paper overcomes this drawback by proposing several most promising parameter setups which are obtained by marginalizing over optimal input points. The pool of such independent starting sets enables the parallel execution of the original simulation code.

Our approach to the prediction of function values by the Gaussian process method was already introduced at MaxEnt 2013 [Preuss and von Toussaint(2014)], which followed in notation and apart from small amendments the book of Rasmussen & Williams [Rasmussen and Williams(2006)]. Therefore, we restrict this introduction only to those formulas needed for the understanding of the new material presented in the chapters below.

Roughly speaking, the Gaussian process method is nothing but extending the solution of the Bayesian linear regression problem with Gaussian distributed noise and a Gaussian prior for the parameters to basically infinite number of dimensions with suitable basis functions. Given is matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$ consisting of N input data vectors $\boldsymbol{x}_i$ of dimension N_{dim}. The target data $\boldsymbol{y} = (y_1, ..., y_N)^T$ is blurred by Gaussian noise of variance $\boldsymbol{\Delta}_{ij} = \sigma_{d_i}^2 \delta_{ij}$. Quantity of interest is the target value f_* at test input vector $\boldsymbol{x}_*$ and generated by a function $f(\boldsymbol{x})$ which shall satisfy $y = f(\boldsymbol{x}) + \epsilon$, with $\langle \epsilon \rangle = 0$ and $\langle \epsilon^2 \rangle = \sigma_d^2$. This states the regression problem for a non-trivial function of unknown shape for which we employ the Gaussian process method. As a statistical process it is fully defined by its covariance function, which is the place where we incorporate all the properties which we would like the (hidden) function describing our problem to have. For the functional form of the covariance we choose a Gaussian type exponent with the negative squared value of the distance between two input data vectors $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$.

$$k(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_f^2 \exp\left\{ -\frac{1}{2} \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\lambda} \right|^2 \right\} \quad . \tag{1}$$

The neighbourhood of the two data vectors should be of relevance for the smoothness of the result, which is mimicked by a length scale λ in the denominator to represent the long range dependence of the two vectors. Moreover, since the Gaussian process method defines a distribution over functions the width of this distribution will have some influence on our result as well. This shall be comprised by the signal variance σ_f^2, however, set to unity if no further information is available. To avoid lengthy formulae, we abbreviate the covariance of the input data as $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the vector of covariances between test input vector and input data as $(\boldsymbol{k}_*)_i = k(\boldsymbol{x}_*, \boldsymbol{x}_i)$. Finally, in addition to the above estimation of the variance of a distinct data point $\sigma_{d_i}^2$

provided by the experimentalist, we consider an overall noise in the data by a variance σ_n^2.

Summing up the analysis from [Preuss and von Toussaint(2014)] the probability distribution for a single function value f_* at test input $\boldsymbol{x}_*$ is

$$p(f_*|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}_*) \propto \mathcal{N}\left(\bar{f}_*, \mathrm{var}(f_*)\right) \quad , \tag{2}$$

with mean

$$\bar{f}_* = \boldsymbol{k}_*^T \left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{\Delta}\right)^{-1} \boldsymbol{y} \quad , \tag{3}$$

and variance

$$\mathrm{var}(f_*) = \boldsymbol{k}(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^T \left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{\Delta}\right)^{-1} \boldsymbol{k}_* \quad . \tag{4}$$

The hyper-parameters $\boldsymbol{\theta}^T = (\lambda, \sigma_f, \sigma_n)$ determine the result of the Gaussian process method. Since we do not know a priori which setting is useful, we marginalize over them numerically by employing the marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathrm{const} - \frac{1}{2}\boldsymbol{y}^T \left[\boldsymbol{K}(\boldsymbol{\theta}) + \sigma_n^2 \boldsymbol{\Delta}\right]^{-1} \boldsymbol{y} - \frac{1}{2} \log \left|\boldsymbol{K}(\boldsymbol{\theta}) + \sigma_n^2 \boldsymbol{\Delta}\right| \quad . \tag{5}$$

The expectation value for the targets $\boldsymbol{f}_*$ at test inputs $\boldsymbol{X}_*$ employs the marginal likelihood and priors for the hyper-parameters from above

$$\langle \boldsymbol{f}_* \rangle = \int \mathrm{d}\boldsymbol{\theta} \; \bar{\boldsymbol{f}}_* \frac{p(\boldsymbol{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int \mathrm{d}\boldsymbol{\theta}' \; p(\boldsymbol{y}|\boldsymbol{\theta}') p(\boldsymbol{\theta}')} \quad , \tag{6}$$

where the fraction term is utilized as the sampling density in Markov chain Monte Carlo. Rescaling of the input data and whitening of the output is performed in order to do the analysis not hampered by large scales or biased from a linear trend. All data has been back-transformed for display.

2 Closed loop optimization scheme

With help of the formulas Eqs. (3) and (4) we can ask for an arbitrary target value and its variance within some region of interest substantiated by existing data. In order to determine the next best points at which to perform an expensive experiment or to run a long term computer code we propose the following autonomous optimization algorithm for sequential parameter selection [Preuss and von Toussaint(2016)].

Since the variance in Eq. (2) depends only on the input $\boldsymbol{X}$ and not on the target data we can immediately evaluate the utility of a further datum without the need to marginalize over the (unknown) target outcome. To achieve this, one has to iterate within a region of interest $\mathcal{I}$ set by the experimentalist over each grid point $\boldsymbol{\xi} \in \mathcal{I}$, which is tentatively handled as being part of the pool of input data vectors. Then for all test inputs $\boldsymbol{x}_* \in \mathcal{I}$ the resulting variances var$'$ have to be determined according to Eq. (2), but with changed $(N+1) \times (N+1)$ covariance matrix $\boldsymbol{K}'$ of the expanded input $\boldsymbol{X}' = \{\boldsymbol{X}, \boldsymbol{\xi}\}$. Summing up the variances var$'$ over all grid points in $\mathcal{I}$ provides a measure for the utility $U(\boldsymbol{\xi})$ of a target data obtained at input vector $\boldsymbol{\xi}$:

$$U(\boldsymbol{\xi}) = - \sum_{\boldsymbol{x}_* \in \mathcal{I}} \mathrm{var}'(f_*) \quad . \tag{7}$$

The minus sign in Eq. (7) reflects the fact that a smaller sum over the variances is connected with a higher utility of the additional datum. The goal is to find $\boldsymbol{\xi}_{\max}$ with the largest utility:

$$\boldsymbol{\xi}_{\max} = \arg \max_{\{\boldsymbol{\xi}\}} U(\boldsymbol{\xi}) \quad . \tag{8}$$

At the obtained $\boldsymbol{\xi}_{\max}$ the next measurement is most informative. If the target data is produced by a computer code, one can set up an autonomously running procedure, which invokes the computer code at $\boldsymbol{\xi}_{\max}$ to add the next target outcome to the data pool, with which the search for the next most informative point is performed. This scheme would be repeated in an iterative manner, till the increase in information from an additional target datum drops below some predefined level or becomes insignificant.

3 Marginalizing test points

The above iterative algorithm proposes further input settings for future evaluations by an automated, but sequential procedure. For applications with long running computer codes it would be desirable to exploit the parallel capabilities of modern multi-processor systems in order to accelerate this process. We propose to create the different onsets for batch operation by invoking a marginalization procedure which simply marginalizes successively over as many test points as batch processors are available. The first processor would be fed with the original problem with the already obtained $N_{\mathrm{start}} = N$ input points. For the second processor we marginalize over the target value f_1 at a first test input vector $\hat{\boldsymbol{x}}_1$ chosen by the estimation criterium of the previous section, i.e. which test point lowers the variance of the full system the most according to Eqs. (7) and (8). The optimal input vector $\boldsymbol{x}_{\mathrm{opt}} = \boldsymbol{\xi}_{\max}$ found is then identified with the input vector $\hat{\boldsymbol{x}}_1$, for which the target value has to marginalized. Then the probability distribution of the next test point f_{1*} is given by the marginalization rule

$$\begin{aligned} p(f_{1*}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{1*}) &= \int \mathrm{d}f_1 p(f_{1*},f_1|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{1*},\hat{\boldsymbol{x}}_1) \\ &= \int \mathrm{d}f_1 p(f_{1*}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{1*},\hat{\boldsymbol{x}}_1,f_1)p(f_1|\boldsymbol{X},\boldsymbol{y},\hat{\boldsymbol{x}}_1) \quad , \end{aligned} \tag{9}$$

where

$$p(f_{1*}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{1*},\hat{\boldsymbol{x}}_1,f_1) \propto \mathcal{N}\left(\bar{f}_{1*},\mathrm{var}(\bar{f}_{1*})|_{f_1}\right) \quad , \tag{10}$$

$$p(f_1|\boldsymbol{X},\boldsymbol{y},\hat{\boldsymbol{x}}_1) \propto \mathcal{N}\left(\bar{f}_1,\mathrm{var}(\bar{f}_1)\right) \quad . \tag{11}$$

We now rewrite Eq. (3) by $\bar{f}_{1*} = \boldsymbol{\kappa}_{1*}^T\boldsymbol{y}_1$, with $\boldsymbol{\kappa}_{1*}^T = \boldsymbol{k}_1^T\left(\boldsymbol{K}+\sigma_n^2\boldsymbol{\Delta}\right)^{-1}$ and $\boldsymbol{y}_1^T = (\boldsymbol{y}^T, f_1)$. For the calculation of Eq. (9) it is expedient to separate the last entry in the κ-vector: $\boldsymbol{\kappa}_{1*}^T = (\hat{\boldsymbol{\kappa}}_{1*}^T, (\boldsymbol{\kappa}_{1*})_{N+1})$. The marginalization integral has to handle Gaussians only and results readily in the predictive distribution for the Gaussian process regression for one marginalized test point f_1

$$p(f_{1*}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{1*}) \propto \mathcal{N}\left(\boldsymbol{\kappa}_{1*}^T\bar{\boldsymbol{y}}_*,\mathrm{var}(\bar{f}_{1*})|_{f_1}\right) \quad , \tag{12}$$

with $\bar{\boldsymbol{y}}_1^T = (\boldsymbol{y}^T, \bar{f}_1)$ and the marginalized variance

$$\mathrm{var}(\bar{f}_{1*})|_{f_1} = \mathrm{var}(\bar{f}_{1*}) + (\boldsymbol{\kappa}_{1*})_{N+1}^2\mathrm{var}(\bar{f}_1) \quad . \tag{13}$$

While the vector multiplication for the mean in Eq. (12) is just the result without marginalization enlarged by the expectation of the marginalized value, the variance has acquired an additional term in Eq. (13) compared to the case without marginalization $\mathrm{var}(\bar{f}_{1*})$. This is reasonable, since the target data pool has increased by an expectation value $\bar{f}_1$ based on the very same data pool, so the lack of new information can only result in a larger variance.

The whole marginalization procedure is extended easily for obtaining the successive points f_2 to f_{N_marg} to be marginalized over. As we intend to use the marginalization procedure for creating a pool of parameter setups to start a long running simulation code in parallel – but for "most valuable" setups only – the order of the successively created setups is of importance.

$$\begin{aligned} \mathrm{var}(\bar{f}_{2*})|_{f_1,f_2} &= \mathrm{var}(\bar{f}_{2*}) + (\boldsymbol{\kappa}_{2*})_{N+1}^2\mathrm{var}(\bar{f}_2)|_{f_1} \quad , \\ &\vdots \\ \mathrm{var}(\bar{f}_{N_\mathrm{marg}*})|_{f_1,\ldots,f_{N_\mathrm{marg}}} &= \mathrm{var}(\bar{f}_{N_\mathrm{marg}*}) + (\boldsymbol{\kappa}_{N_\mathrm{marg}*})_{N+1}^2\mathrm{var}(\bar{f}_{(N_\mathrm{marg}-1)*})|_{f_1,\ldots,f_{N_\mathrm{marg}-1}} \end{aligned} \tag{14}$$

Once the variances are calculated, the largest utility can be obtained by using Eqs. (7) and (8).

4 Validation in one dimension

In order to examine the dependence of the result on the marginalized points we simulate a one-dimensional test case, which is analytically accessible. For an input data set consisting of a single point at x_1 with target value y_1 the expectation value at test point x_* is

$$\bar{f}_* = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2 \sigma_{d_1}^2} \exp\left\{ -\frac{1}{2} \left| \frac{x_1 - x_*}{\lambda} \right|^2 \right\} y_1 \quad . \tag{15}$$

with variance

$$\mathrm{var}(\bar{f}_*) = \sigma_f^2 - \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2 \sigma_{d_1}^2} \exp\left\{ - \left| \frac{x_1 - x_*}{\lambda} \right|^2 \right\} y_1 \quad . \tag{16}$$

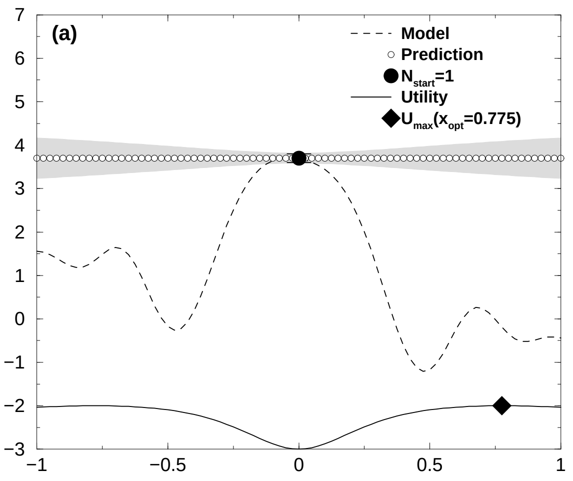
This is plotted in Fig. 1a with $x_1 = 0$. The y-value is taken from the sinusoidal model shown (a cos-function with decreasing amplitude), however subjected to whitening, which results in this simple case of one point to the assignment of a zero value. The back-transformation to the scale of the input target value leads to the horizontal line titled "Prediction" (marked by small white circles). As can be seen from Eq. (16), the variance increases with the distance from the data point. The utility according to Eq. (7) was scaled and normalized to fit within the range $[-2 : -3]$ of each graph. It produces the result of $x_{\mathrm{opt}} = 0.775$ for the next best point to measure at. Without marginalization this would be the input datum returning the results of Fig. 1b for our simple one dimensional model. After ten iterations the prediction is – within the gray uncertainty region – almost pinned down to the model generating function (see Fig. 1c).

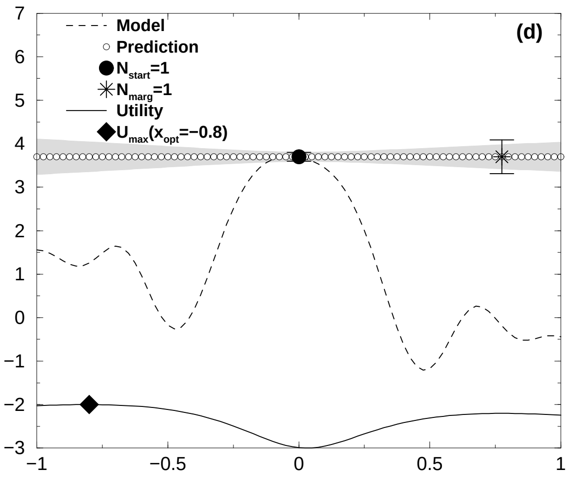
To run the whole process on a second processor one would take the first optimal point at $\hat{x}_1 = 0.775$ for the marginalization step, i.e. its target value f_1 does not enter the data base. The prediction can be done analytically as well to give

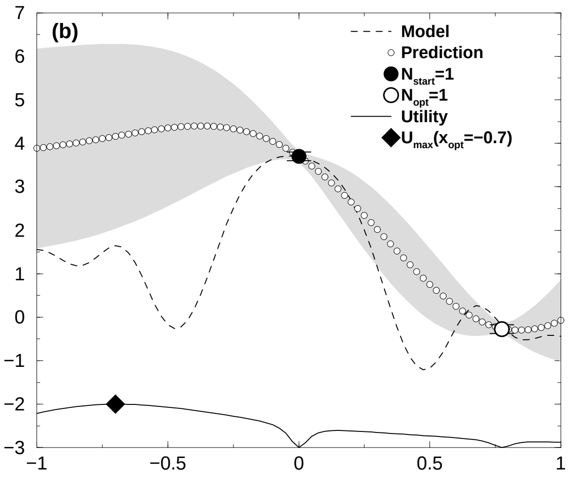
$$\bar{f}_{1*} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2 \sigma_{d_1}^2} \exp\left\{ -\frac{1}{2} \left| \frac{x_1 - x_{1*}}{\lambda} \right|^2 \right\} y_1 + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2 \mathrm{var}(\bar{f}_*)} \exp\left\{ -\frac{1}{2} \left| \frac{x_* - x_{1*}}{\lambda} \right|^2 \right\} \bar{f}_* \quad , \tag{17}$$

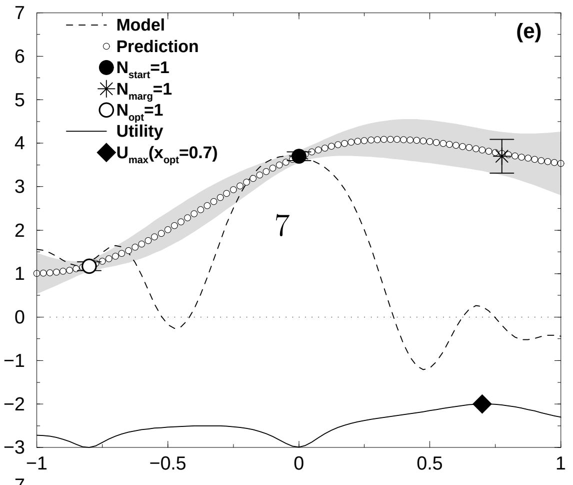
with variance

$$\begin{aligned} \mathrm{var}(\bar{f}_{1*}) = \sigma_f^2 - \Bigg[& \quad \left(\sigma_f^2 + \sigma_n^2 \mathrm{var}(\bar{f}_*)\right) \exp\left\{ - \left| \frac{x_1 - x_{1*}}{\lambda} \right|^2 \right\} \\ & - \quad 2\sigma_f^2 \exp\left\{ - \frac{(x_* - x_{1*})^2 + (x_1 - x_*)^2 + (x_1 - x_{1*})^2}{2\lambda^2} \right\} \end{aligned}$$


(a)
Model
Prediction
N_{start}=1
Utility
$U_{max}(x_{opt}$=0.775)


(d)
Model
Prediction
N_{start}=1
N_{marg}=1
Utility
$U_{max}(x_{opt}$=−0.8)


(b)
Model
Prediction
N_{start}=1
N_{opt}=1
Utility
$U_{max}(x_{opt}$=−0.7)


(e)
Model
Prediction
N_{start}=1
N_{marg}=1
N_{opt}=1
Utility
$U_{max}(x_{opt}$=0.7)

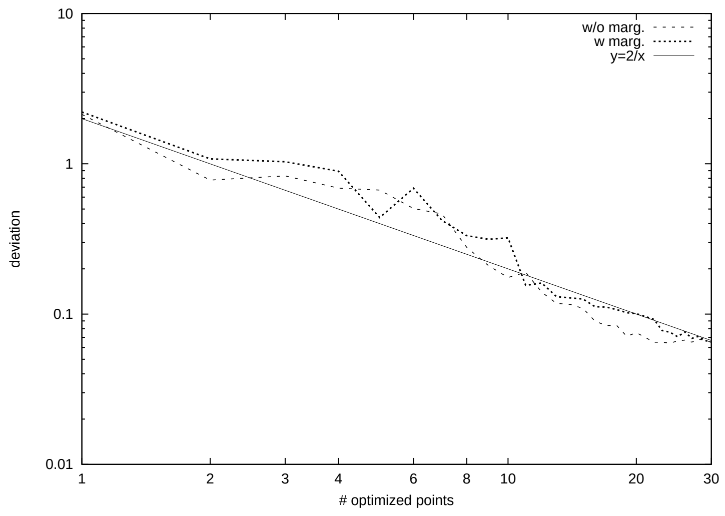


Figure 2: Deviation of the two batch runs with and without one marginalized input value from the exact model.

$$+ \qquad \left(\sigma_f^2 + \sigma_n^2 \sigma_{d_1}^2\right) \exp\left\{-\left|\frac{x_* - x_{1*}}{\lambda}\right|^2\right\}\Bigg]$$

$$/ \qquad \left[\left(\sigma_f^2 + \sigma_n^2 \sigma_{d_1}^2\right)\left(\sigma_f^2 + \sigma_n^2 \mathrm{var}(\bar{f}_*)\right)/\sigma_f^4 - \exp\left\{-\left|\frac{x_* - x_1}{\lambda}\right|^2\right\}\right] \quad (18)$$

As can be seen in Fig. 1d the marginalized target point does not affect the prediction much (both y_1 and $\bar{f}_*$ are zero after whitening and yield a zero line according to Eq. (17)). Moreover, the standard deviation of the marginalized point, $\sqrt{\mathrm{var}(\bar{f}_{1*})} = 0.3895$ is nearly four times higher than the inputted measurement uncertainty of $\sigma_{d_1} = 0.1$, which emphasizes the fact that the marginalized point is of less importance. The utility in Fig. 1d gives rise for the next best point for a measurement at $x_{\mathrm{opt},1} = -0.8$, but after obtaining the target value, already the next utility in Fig. 1e has its maximum at $x_{\mathrm{opt},2} = 0.7$ close to the position of the marginalized point $x_{\mathrm{marg}} = 0.775$. This, together with further points obtained by the optimization procedure, eventually concludes to a close similarity of the predicted curve with the model (see Fig. 1f). The marginalized point is of no importance, as should be the case,

That the marginalization procedure does not affect the result based on the increasing input data pool is supported by Fig. 2 depicting the deviation of the prediction from the model as a function of the number of optimized points in the input data. Both batch runs – with and without marginalization – show the same descent.

5 Algorithm for computer simulation

Eventually, we briefly describe an algorithm for a computer simulation employing the batch approach introduced. The field of application will be the prediction of particle transport and plasma-wall interaction in the scrape-off layer in fusion plasma experiments. The usual running time for obtaining the outcome for one set of parameters is in the order of months, so the particular setups have to be chosen well. Up to now a data base with the outcome for about 1500 parameter settings has been established [Coster(2014)].

1. Compose input data vector from data base
2. Set up batch run with N_{proc} processors
3. Processor #1: code running without any marginalized point
   Processor #$(i + 1)$: code running for i marginalized points
4. Return outcome, i.e. N_{proc} most promising parameter settings for the long running simulation code ready for batch execution.

6 Conclusion

A marginalization procedure was proposed to obtain different starting conditions for batch execution of Gaussian processes in order to exploit parallel computing power. By investigating a one-dimensional test case the implementation of the procedure (algorithm, computer program, MCMC results) was validated with analytic calculations. The marginalized input point was shown to become insignificant for the final result.

7 Acknowledgement

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[Preuss and von Toussaint(2014)] R. Preuss, and U. von Toussaint, "Prediction of Plasma Simulation Data with the Gaussian Process Method," in *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, edited by R. Niven, AIP Publishing, Melville, NY, 2014, vol. 1636, p. 118.

[Rasmussen and Williams(2006)] C. Rasmussen, and C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, 2006.

[Preuss and von Toussaint(2016)] R. Preuss, and U. von Toussaint, *Fusion Sci. Tech.* **69**, 605–610 (2016).

[Coster(2014)] D. Coster (2014), private communication.

8 Appendix

Notation table

N	number of input data vectors
N_{dim}	number of elements in the input data vector
N_{marg}	number of marginalized test data
$\boldsymbol{x}_*$	test input vector
$\hat{\boldsymbol{x}}_1$	first test input vector, for which the target value is marginalized
$\boldsymbol{x}_{\text{opt}}$	test input vector found by the utility criterium
$\boldsymbol{x}_{1*}$	test input vector after first marginalized test input vector was found
$\boldsymbol{x}_i = (x_{i1}, ..., x_{iN_{\text{dim}}})$	i-th input data vector
$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$	$N \times N_{dim}$ matrix with input data vectors as columns
$\boldsymbol{X}' = \{\boldsymbol{X}, \boldsymbol{\xi}\}$	matrix of the input data vectors expanded by the vector of grid poin
$\boldsymbol{\xi}$	vector of grid points within region of interest $\mathcal{I}$
$\boldsymbol{\xi}_{\text{max}}$	grid point with largest utility
f_*	target value at test input vector $\boldsymbol{x}_*$
$f(\boldsymbol{x})$	function of input data to describe target data
f_1	first target value, to be marginalized
f_{1*}	target value at test point, obtained after marginalization of a first ta
$\boldsymbol{y} = (y_1, ..., y_N)^T$	vector of the N target data
ϵ	uncertainty of the target data
$\sigma_{d_i}^2$	variance of the i-th target data
$\boldsymbol{\Delta}_{ij} = \sigma_{d_i}^2 \delta_{ij}$	ij-th element of the $N \times N$ matrix of the variances of target data
λ	length scale to set up the notion of distance between input data vect
σ_f^2	signal variance of the distribution over functions f
σ_n^2	overall noise in the data
$\boldsymbol{\theta} = (\lambda, \sigma_f, \sigma_n)$	vector of the hyperparameters
$k(\boldsymbol{x}_p, \boldsymbol{x}_q)$	covariance of two input data vectors
$(\boldsymbol{k}_*)_i = k(\boldsymbol{x}_*, \boldsymbol{x}_i)$	short notation for the i-th element of the vector of covariances betwe input vector and input data vector
$\boldsymbol{K}_{pq} = k(\boldsymbol{x}_p, \boldsymbol{x}_q)$	pq-th element of the $N \times N$ covariance matrix of the input data vect
$\boldsymbol{K}'$	$(N+1) \times (N+1)$ covariance matrix of the expanded input $\boldsymbol{X}'$
$\mathcal{I}$	region of interest to run Gaussian processes
$U(\boldsymbol{\xi})$	utility of a target data obtained at input vector ξ