Prediction of SOLPS data employing Gaussian Processes

R. Preuss, U. von Toussaint

Max-Planck-Institute for Plasma Physics, 85748 Garching, Germany

Introduction

The simulation of plasma-wall interactions of fusion plasmas is extremely costly in computer power and time – the running time for a single parameter setting is easily in the order of weeks or months, not to mention the expenditure for parametric studies. Still there exists a small choice of data bases with several hundred entries in different parameter regimes acquired by various research groups. Based on these already gathered results the idea is to deploy an emulator capable of predicting dense answers in regions in data space with sufficient support. In the absence of a concise descriptive model this formulates a regression problem for a non-trivial function of unknown shape. Particularly useful for such tasks with various numbers of dimensions for input data or target function is the Gaussian process method. Resulting function values are not confined to a subclass in function space but constitute a dense representation of any uniformly continuous function. The method has been appreciated much in the fields of neural networks and machine learning [1, 2, 3]. Following in notation the book of Rasmussen & Williams [4] we utilize the method within a probability-theoretical (Bayesian) framework. It has to be kept in mind that the prediction gets weak whenever the data support becomes sparse or in the vicinity of phase transitions and for the description thereof. Most other function parameterising methods like e.g. neural networks or polynomial fitting tend rapidly to unreasonable results outside of the support region. Not so Gaussian processes, which show an uninformative, smooth behavior with largely increased uncertainties. The thereby easily attainable identification of regions of comparable high uncertainties point the way where the next simulation runs have to be accomplished to supplement the data base.

Prediction of function values

Given n input data vectors $\vec{x}_i$ of dimension N_{dim} (with matrix $\underline{\underline{X}} = (\vec{x}_1, \vec{x}_2, ..., \vec{x}_n)$) and corresponding target data $\vec{y} = (y_1, ..., y_n)^T$ blurred by Gaussian noise of variance σ_d^2 the quested quantity is the target value f_* at test input vector $\vec{x}_*$. The later would be generated by a function $f(\vec{x})$, with $y = f(\vec{x}) + \varepsilon$, where $\langle \varepsilon \rangle = 0$ and $\langle \varepsilon^2 \rangle = \sigma_d^2$. For being completely ignorant about a model describing function our Ansatz is to employ the Gaussian process method, with which any uniformly continuous function may be represented. As a statistical process it is fully defined by its covariance function and called Gaussian, because any collection of random variables

produced by this process has a Gaussian distribution.

The choice of the covariance function is decisive for the inference we want to apply. It is the place where we incorporate all the properties which we would like our (hidden) problem describing function to have in order to influence the result. For example, the neighbourhood of two input data vectors $\vec{x}_p$ and $\vec{x}_q$ should be of relevance for the smoothness of the result. This shall be expressed by a length scale λ which represents the long range dependence of the two vectors. For the covariance function itself we employ a Gaussian type exponent with the negative squared value of the distance between two vectors $\vec{x}_p$ and $\vec{x}_q$

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

σ_f^2 is the signal variance and apriori set to one, if we are ignorant about this value. To avoid lengthy formulae, we abbreviate the covariance matrix of the input data as $(\underline{\underline{K}})_{ij} = k(\vec{x}_i,\vec{x}_j)$ and the vector of covariances between test point and input data as $(\vec{k}_*)_i = k(\vec{x}_*,\vec{x}_i)$.

Moreover, we consider the degree of information which the data possesses by an overall variance σ_n^2 accounting that the data are noisy and – more detailed – $(\sigma_d)_i$ for the uncertainty estimation of a single data point y_i provided by the experimentalist. It can be shown [4] that for given λ, σ_f and σ_n the probability distribution for a single function value f_* is

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

with mean and variance

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

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

$\underline{\underline{\Delta}}$ is a matrix with the variances $\vec{\sigma}_d^2$ of the input data on its diagonal and zero otherwise. If no uncertainties of the input data are provided, $\underline{\underline{\Delta}}$ is set to the identity matrix.

Marginalizing the hyper-parameters

The hyper-parameters $\vec{\theta} = (\lambda, \sigma_f, \sigma_n)^T$ determine the result of the Gaussian process method. Since we do not know a priori, which setting is useful, we marginalize over them later on in order to get the target values $\vec{f}_*$ for test inputs $\vec{X}_*$. Their expectation values are

$$\langle\vec{\theta}\rangle = \frac{\int \mathrm{d}\vec{\theta}\ \vec{\theta} p(\vec{\theta}|\vec{y})}{\int \mathrm{d}\vec{\theta}\ p(\vec{\theta}|\vec{y})} = \frac{\int \mathrm{d}\vec{\theta}\ \vec{\theta} p(\vec{y}|\vec{\theta}) p(\vec{\theta})}{\int \mathrm{d}\vec{\theta}\ p(\vec{y}|\vec{\theta}) p(\vec{\theta})} \quad . \tag{5}$$

Gaussian priors are employed for the hyper-parameters with mean and variance one but constrained to be positive,

$$p(\theta_i) \sim \mathcal{N}(1,1)\ \ \forall\ \ \theta_i \geq 0 \quad \text{and} \quad p(\theta_i) = 0\ \ \text{otherwise} \quad . \tag{6}$$

The marginal likelihood $p(\vec{y}|\vec{\theta})$ is obtained by

$$p(\vec{y}|\vec{\theta}) = \int \mathrm{d}\vec{f}\; p(\vec{y}|\vec{f},\vec{\theta}) p(\vec{f}|\vec{\theta}) \quad . \tag{7}$$

As we deal with the Gaussian process the probability functions are of Gaussian type, with the likelihood as $p(\vec{y}|\vec{f},\vec{\theta}) \sim \mathcal{N}(\vec{f},\sigma_n\underline{\underline{\Delta}})$ and the prior for $\vec{f}$ as $p(\vec{f}|\vec{\theta}) \sim \mathcal{N}(\vec{0},\underline{\underline{K}})$ [4]. Thus the integration in Eq. (7) yields

$$\log p(\vec{y}|\vec{\theta}) \sim -\frac{1}{2}\vec{y}^T \left[\underline{\underline{K}}(\vec{\theta}) + \sigma_n^2\underline{\underline{\Delta}}\right]^{-1}\vec{y} - \frac{1}{2}\log\left|\underline{\underline{K}}(\vec{\theta}) + \sigma_n^2\underline{\underline{\Delta}}\right| \quad . \tag{8}$$

The expectation value for the target f_* at test input $\vec{x}_*$ employs the marginal likelihood and priors for the hyper-parameters from above

$$\langle \vec{f}_* \rangle = \int \mathrm{d}\vec{\theta}\; \bar{f}_* \frac{p(\vec{y}|\vec{\theta})p(\vec{\theta})}{\int \mathrm{d}\vec{\theta}'\; p(\vec{y}|\vec{\theta}')p(\vec{\theta}')} \quad , \tag{9}$$

where the fraction constitutes the sampling density for Markov chain Monte Carlo evaluation of the integral.

SOLPS data example

The power of the Gaussian process method is its straightforward applicability in any number of dimensions for input data or target function. This is of special benefit in spaces of various (fusion plasma) input and output parameters, if the number of source data is already sufficient for reliable inferences. A particular computationally expensive area is that of predicting the outcome of particle transport and plasma-wall interaction in the scrape-off layer in fusion plasma experiments. Here the theoretically acquired results are obtained by the interplay of two sophisticated codes either describing the plasma solving a fluid equation or the transport of neutrals by a Monte-Carlo method. The run for a single parameter setting is in the order of weeks, sometimes even several months on the fastest many-core computers available. A data base of 1500 parameter settings will be the platform we intend

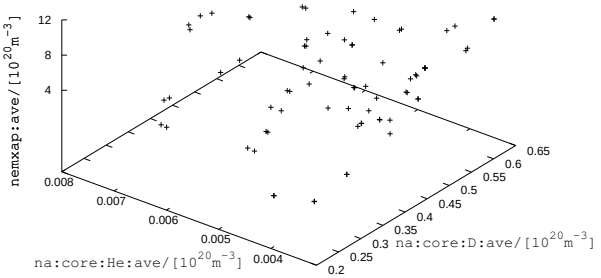

Figure 1: *Scrape-off layer plasma simulation (SOLPS) data for the maximum electron density at the outboard divertor as the target function and the core densities of deuterium and helium as two-dimensional input. Only data from the SOLPS database with ELM set to false and densities for deuterium above* $10^{19}/m^3$ *and helium above* $3.1\times10^{17}/m^3$ *are considered making a data pool of 76 entries.*

to start from to make inferences about outcomes within the ranges of the acquired data. To keep it instructive we restrict ourselves in this paper to the two-dimensional space for the input data (the core densities of deuterium `na:core:D:ave` and helium `na:core:He:ave`) and one-dimensional target function (maximum electron density, outboard divertor `nemxap:ave`). Further restrictions on the data set caused by physics considerations (e.g. density constraints) leave a number of 76 input data vectors. Fig. 2 depicts the outcome of the Gaussian process method. The coloured hyper-plane is the expectation value of the target function. The uncertainties of the predictions are displayed in normalized colour code and indicate (in yellow) those parameter regions in input space where further (expensive) computations should take place to enforce the reliability of the outcome. This provides a feasible strategy towards an autonomous optimization algorithm.

Acknowledgment

We would like to thank D. Coster for providing the SOLPS-simulation data (partly) before publication. This project has received funding from the Euratom research and training programme 2014-2018.

References

[1] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, (2012).

[2] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, (1996).

[3] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, (2003).

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

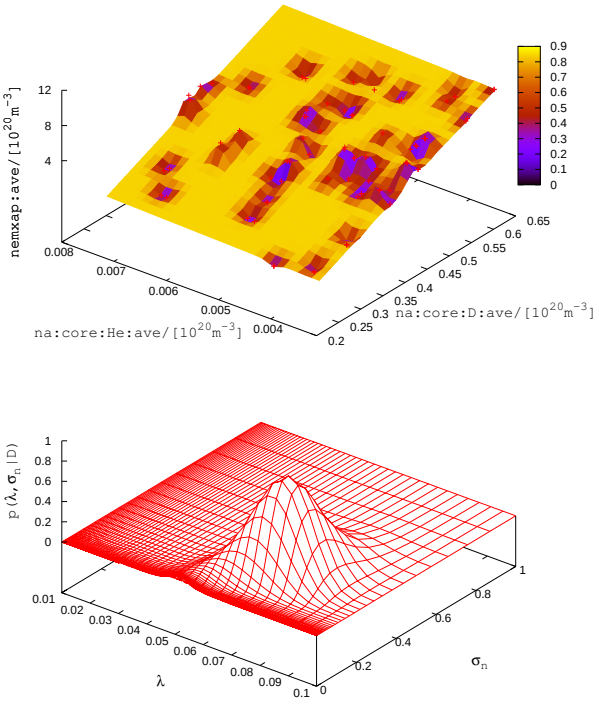

Figure 2: *Top: Predictive mean for the maximum electron density data of Fig. 1. For violet (darker) points one is pretty sure about the prediction. Further experiments should take place for parameter settings at yellow (lighter) areas. Bottom: marginal likelihood for λ and σ_n. The pronounced peak ensures a well-behaved calculation of the expectation value integral.*