lomonosov Moscow State University
faculty of computational mathematics and cybernatics
chair of mathematical methods of forecasting

Izmailov Pavel

Gaussian processes for Machine Learning

COURSE WORK

Scientific advisors:
D. P. Vetrov
D. A. Kropotov

Moscow, 2016

1 Introduction

Gaussian processes provide an elegant and effective approach to learning in kernel machines. This approach leads to a highly interpretable model and allows using the bayesian framework for model adaptation and incorporating the prior knowledge about the problem.

1.1 Gaussian process definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

We will only consider processes, that take place in a finite dimensional real space $\mathbb{R}^d$. In this case, f is a Gaussian process, if for any k, for any $t_1, t_2, \ldots, t_k \in \mathbb{R}^d$ the joint distribution

$$(f(t_1), f(t_2), \ldots, f(t_k))^T \sim \mathcal{N}(m_t, K_t)$$

for some m_t and K_t.

The mean m_t of this distribution is defined by the mean function $m : \mathbb{R}^d \to \mathbb{R}$:

$$m_t = (m(t_1), m(t_2), \ldots, m(t_k))^T.$$

Similarly, the covariance matrix K_t is defined by the covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$:

$$K_t = \begin{pmatrix} k(t_1,t_1) & k(t_1,t_2) & \ldots & k(t_1,t_n) \\ k(t_2,t_1) & k(t_2,t_2) & \ldots & k(t_2,t_n) \\ \ldots & \ldots & \ldots & \ldots \\ k(t_n,t_1) & k(t_n,t_2) & \ldots & k(t_n,t_n) \end{pmatrix}.$$

It's straightforward then, that a Gaussian process is completely defined by it's mean and covariance functions. We will use the following notation.

$$f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$$

means that f is a Gaussian process with mean function m and covariance function k. While the mean function can be an arbitrary real-valued function, the covariance function has to be a kernel, so that the covariance matrices it implies are symmetric and positive definite.

In the fig. 1 an example of a one-dimensional Gaussian process is provided. The darker blue line is the mean of the process, and the lighter blue region is the 3σ-region, drawn at each point.

1.2 Gaussian process regression

Consider the regression problem. We have a dataset $\{(x_i, f_i)|i = 1, \ldots, n\}$, which is considered to be generated from an unknown Gaussian process $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, let $x \in \mathbb{R}^d$. We will denote the matrix comprised of points $x_1, \ldots, x_n$ by $X \in \mathbb{R}^{n \times d}$ and the vector of corresponding target values $f_1, ..., f_n$ by $f \in \mathbb{R}^n$. We want to predict the values $f_* \in \mathbb{R}^l$ of the unknown process at a set of other l points $X_* \in \mathbb{R}^{l \times d}$.

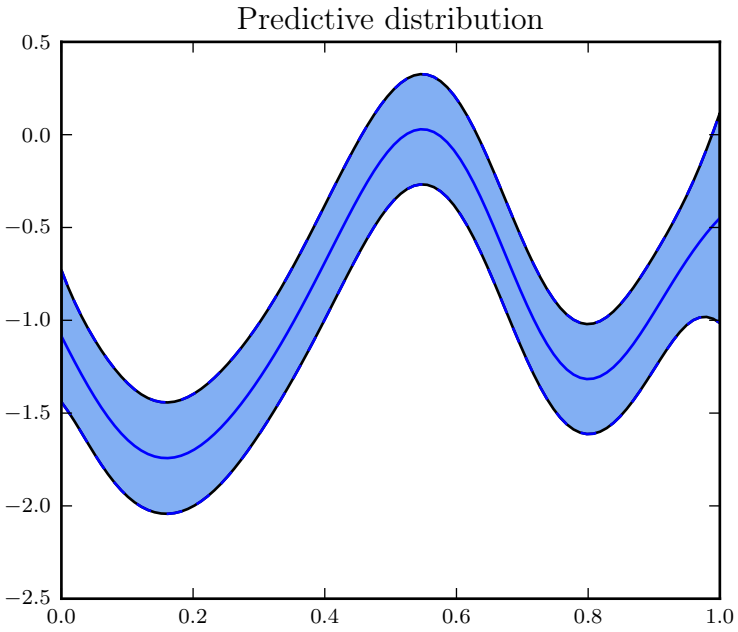

Fig. 1: One-dimensional Gaussian processes

We will also consider the case, when we can't directly observe the values f of the process at points X. Instead, we will use their noisy versions y, which are distributed as

$$y \sim \mathcal{N}(f, \sigma_n^2 I),$$

for some noise variance $\sigma_n \in \mathbb{R}$. The graphical model for this setting is provided in fig. 2.

We will use the following notaion. We will denote the matrix, comprised of pairwise values, of covariance functions on two sets of points $A = (a_1, \ldots, a_n)^T \in \mathbb{R}^{n \times d}$ and $B = (b_1, \ldots, b_m)^T \in \mathbb{R}^{m \times d}$ by

$$K(A, B) = \begin{pmatrix} k(a_1, b_1) & k(a_1, b_2) & \ldots & k(a_1, b_m) \\ k(a_2, b_1) & k(a_2, b_2) & \ldots & k(a_2, b_m) \\ \ldots & \ldots & \ldots & \ldots \\ k(a_n, b_1) & k(a_n, b_2) & \ldots & k(a_n, b_m) \end{pmatrix} \in \mathbb{R}^{n \times m}.$$

Then, the joint distribution of f and f_* is given by

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right).$$

As y is obtained by adding a normally distributed noise to f, the joint distribution

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right).$$

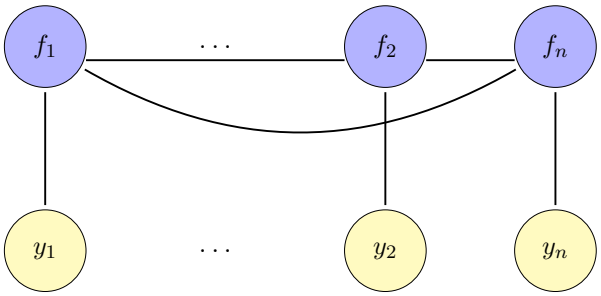

Fig. 2: The graphical model for Gaussian process regression and classification

Conditioning this distribution, we obtain the predictive

$$f_*|X_*, X, f \sim \mathcal{N}(\hat{m}, \hat{K}),$$

where

$$\mathbb{E}[f_*|f] = \hat{m} = K(X_*, X)K(X, X)^{-1}f,$$
$$\text{cov}(f_*|f) = \hat{K} = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*).$$

Thus, the complexity of obtaining the predictive distribution, is determined by the complexity of inverting the n by n matrix $K(X, X)$ and thus scales as $\mathcal{O}(n^3)$.

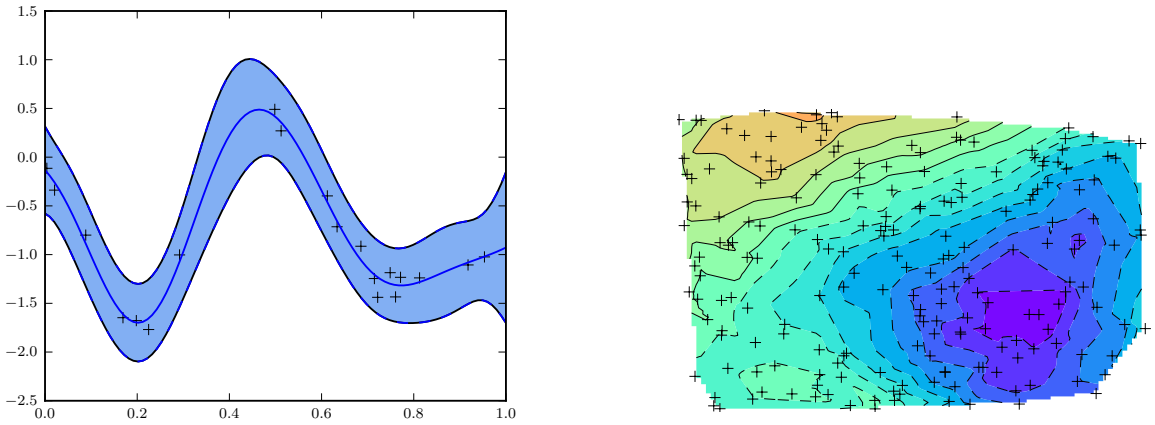

Fig. 3: One and two-dimensional Gaussian processes, reconstructed from data

In fig. 3 you can see the examples of one and two-dimensional Gaussian processes, reconstructed from the data. The data points are shown by black '+' signs.

For more detailed description of Gaussian process regression see [1].

1.3 Gaussian process classification

Now we will apply the Gaussian processes to the binary classification problem, which can be described as follows. We have a dataset $\{(x_i, y_i)|i = 1, \ldots, n\}$, where $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$.

We want to predict the probabilities of new datapoints x_* belonging to positive class.

We will consider the following model. We will introduce a latent function $f : \mathbb{R}^d \to \mathbb{R}$ and put a zero-mean GP prior over it.

$$f \sim \mathcal{GP}(0, k(\cdot, \cdot)).$$

We will then consider the probability of the object x_* belonging to positive class, to be equal to $\sigma(f(x_*))$ for the chosen sigmoid function σ.

$$p(y_* = +1|x_*) = \sigma(f(x_*)).$$

Note, that the graphical for this model is exactly the same, as for the regression problem and is given in fig. 2.

We will use the logistic function $\sigma(z) = (1 + \exp(-z))^{-1}$, however one can use other sigmoid functions as well.

Now inference can be done in two steps. First, for the new data point x_* we should find the conditional distribution of the value of the latent process f at the new data point x_*. This can be computed as follows

$$p(f_*|X, y, x_*) = \int p(f_*|X, x_*, f)p(f|X, y)df. \tag{1}$$

Now, the probability of the positive class is given by marginalizing over the latent variable f_*.

$$p(y_* = +1|X, y, x_*) = \int \sigma(f_*)p(f_*|X, y, x_*)df_*. \tag{2}$$

Unfortunantely, both the integrals in (1) and (2) are intractable. Thus, we have to use integral-approximation techniques to estimate the predictive distribution.

For example, one can use Laplace approximation method, which builds a Gaussian approximation $q(f|X, y)$ to the true posterior $p(f|X, y)$. This approximation is obtained, by performing the Taylor expansion of the function $\log p(f|X, y)$ around it's maximum $\hat{f}$.

Substituting this Gaussian approximation back into (1) and (2), we obtain tractable integrals, and can compute the predictive distribution in a closed form. The more detailed derivation of this algorithm and another algorithm, based on Expectation Propagation can be found in [1].

We will also describe another method for GP-classification below.

Computational complexity of computing the predictive distribution for this method scales as $\mathcal{O}(n^3)$.

1.4 Model adaptation

In the previous two sections, we described, how to fit a Gaussian process to the data in the regression and in the classification problem. However, we only considered the Gaussian processes with fixed covariance functions. This model can be rather limiting.

Most of the popular covariance functions have a set of parameters, which we will refer to as covariance (or kernel) hyper-parameters. For example, the squared exponential covariance function

$$k_{SE}(x, x') = \sigma^2 \exp -\frac{||x - x'||^2}{l^2}$$

has two parameters — variance σ and length-scale l. An example of a more complicated popular covariance function is the Matern function, given by

$$k_{Matern}(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}||x - x'||}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{||x - x'||} l \right),$$

with two positive parameters ν and l. Here K_{ν} is a modified Bessel function.

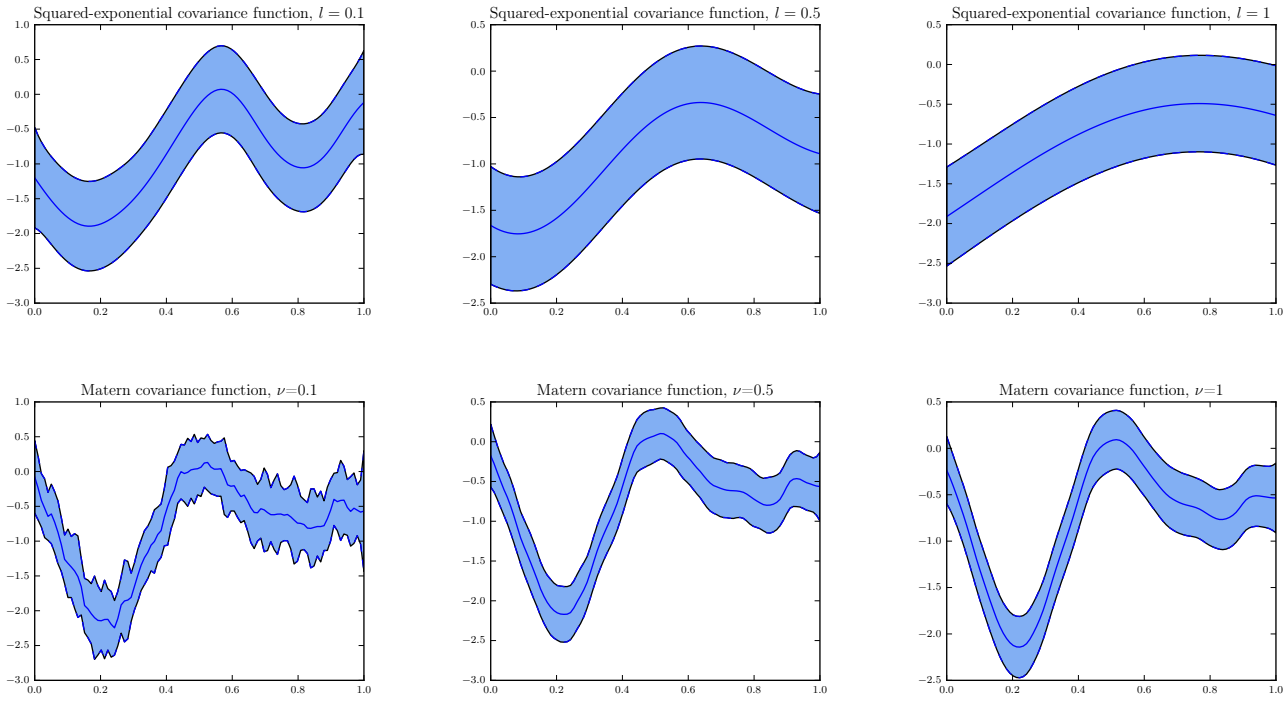

Fig. 4: Gaussian processes with squared-exponential and Matern covariance functions, reconstructed from the same data for different values of hyper-parameters

In fig. 4 you can see the predictive distributions of the Gaussian-process regression for the same dataset for different values of kernel hyper-parameters of the squared exponential and Matern covariance functions. It can be seen from these plots, that in order to get a good model for the data, one should find a good set of kernel hyper-parameters.

Bayesian paradigm provides a way of tuning the kernel hyper-parameters of the GP-model through maximizization of the evidence, or marginal likelihood of the model. Marginal

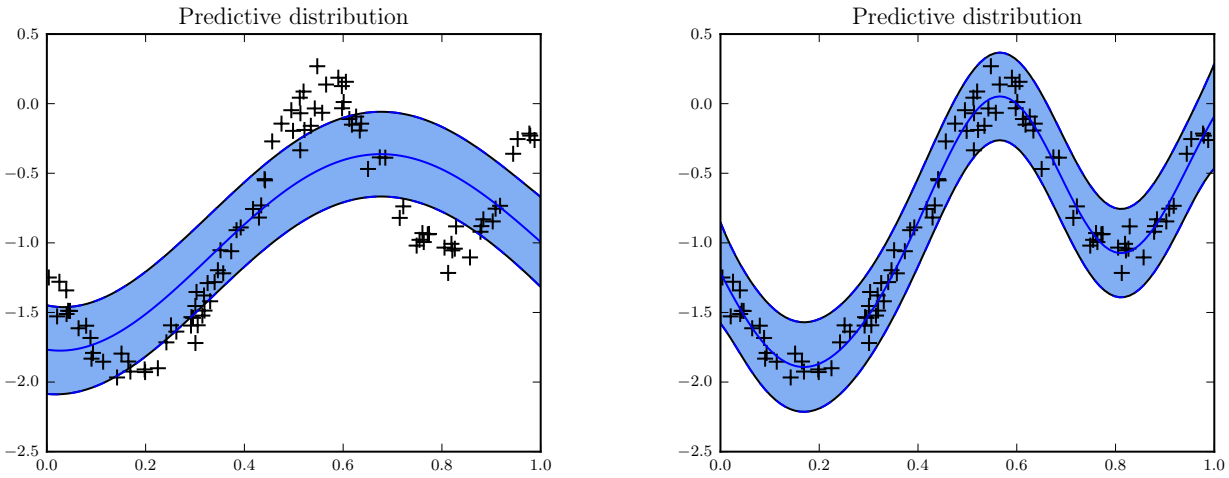

Fig. 5: Predictive distribution before and after hyper-parameter adaptation

likelihood is given by

$$p(y|X) = \int p(y|f, X)p(f|X)df,$$

which is the likelihood, marginalized over the latent values f of the underlying process.

For the GP-regression the marginal likelihood can be computed in closed form and is given by

$$\log p(y|X) = -\frac{1}{2}y^T(K + \sigma_n^2 I)^{-1}y - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi. \tag{3}$$

For the laplace approximation method, the marginal likelihhod is also available in the closed form.

Thus, for the regression problem, we should first maximize the evidence, given by (3) with respect to covariance hyper-parameters and then use the predictive distribution, derived in section 1.2, to make predictions for the new data points.

For the Laplace approximation method for the classification problem, the procedure is slightly more complicated and is described in [1].

Fig. 5 provides an example of the GP-regression predictive distribution for the same dataset before and after tuning the kernel hyper-parameters. It can be seen from the plots, that the model with tuned hyper-parameters, describes the data much better.

Literature

[1] Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. *MIT Press*.

[2] Titsias M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In: *International Conference on Artificial Intelligence and Statistics*, pp. 567–574.

[3] Hensman J., Fusi N., Lawrence D. (2013). Gaussian Processes for Big Data. In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.

[4] Hensman J., Matthews G., Ghahramani Z. (2015). Scalable Variational Gaussian Process Classification. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*.