# Learning from data: Gaussian processes

**Christian Forssén**

Department of Physics, Chalmers University of Technology, Sweden

Oct 7, 2019

## 0.1 Inference using Gaussian processes

Assume that there is a set of input vectors with independent, predictor, variables

$$\boldsymbol{X}_N \equiv \{\boldsymbol{x}^{(n)}\}_{n=1}^N$$

and a set of target values

$$\boldsymbol{t}_N \equiv \{t^{(n)}\}_{n=1}^N.$$

- Note that we will use the symbol $t$ to denote the target, or response, variables in the context of Gaussian Processes.
- Furthermore, we will use the subscript $N$ to denote a vector of $N$ vectors (or scalars): $\boldsymbol{X}_N$ $(\boldsymbol{t}_N)$
- While a single instance $i$ is denoted by a superscript: $\boldsymbol{x}^{(i)}$ $(t^{(i)})$.

We will consider two different *inference problems*:

1. The prediction of *new target* $t^{(N+1)}$ given a new input $\boldsymbol{x}^{(N+1)}$
2. The inference of a *function* $y(\boldsymbol{x})$ from the data.

The former can be expressed with the pdf

$$p\left(t^{(N+1)}|\boldsymbol{t}_N, \boldsymbol{X}_N, \boldsymbol{x}^{(N+1)}\right)$$

while the latter can be written using Bayes' formula (in these notes we will not be including information $I$ explicitly in the conditional probabilities)

$$p\left(y(\boldsymbol{x})|\boldsymbol{t}_N, \boldsymbol{X}_N\right) = \frac{p\left(\boldsymbol{t}_N|y(\boldsymbol{x}), \boldsymbol{X}_N\right) p\left(y(\boldsymbol{x})\right)}{p\left(\boldsymbol{t}_N|\boldsymbol{X}_N\right)}$$

The inference of a function will obviously also allow to make predictions for new targets. However, we will need to consider in particular the second term in the numerator, which is the **prior** distribution on functions assumed in the model.

- This prior is implicit in parametric models with priors on the parameters.
- The idea of Gaussian process modeling is to put a prior directly on the **space of functions** without parameterizing $y(\boldsymbol{x})$.
- A Gaussian process can be thought of as a generalization of a Gaussian distribution over a finite vector space to a **function space of infinite dimension**.
- Just as a Gaussian distribution is specified by its mean and covariance matrix, a Gaussian process is specified by a **mean and covariance function**.

**Gaussian process**

A Gaussian process is a stochastic process (a collection of random variables indexed by time or space), such that every finite collection of those random variables has a multivariate normal distribution

**References:**

1. Gaussian Processes for Machine Learning, Carl Edward Rasmussen and Chris Williams, the MIT Press, 2006, online version.
2. GPy: a Gaussian Process (GP) framework written in python, from the Sheffield machine learning group.

## 0.2 Parametric approach

Let us express $y(\boldsymbol{x})$ in terms of a model function $y(\boldsymbol{x};\boldsymbol{\theta})$ that depends on a vector of model parameters $\boldsymbol{\theta}$.

For example, using a set of basis functions $\left\{\phi^{(h)}(\boldsymbol{x})\right\}_{h=1}^{H}$ with linear weights $\boldsymbol{\theta}_H$ we have

$$y(\boldsymbol{x},\boldsymbol{\theta}) = \sum_{h=1}^{H} \theta^{(h)}\phi^{(h)}(\boldsymbol{x})$$

**Notice**

The basis functions can be non-linear such as Gaussians (aka *radial basis functions*)

$$\phi^{(h)}(\boldsymbol{x}) = \exp\left[-\frac{\left(\boldsymbol{x}-\boldsymbol{c}^{(h)}\right)^2}{2(\sigma^{(h)})^2}\right].$$

Still, this constitutes a linear model since $y(\boldsymbol{x}, \boldsymbol{\theta})$ depends linearly on the parameters $\boldsymbol{\theta}$.

The inference of model parameters should be a well-known problem by now. We state it in terms of Bayes theorem

$$p\left(\boldsymbol{\theta}|\boldsymbol{t}_N, \boldsymbol{X}_N\right) = \frac{p\left(\boldsymbol{t}_N|\boldsymbol{\theta}, \boldsymbol{X}_N\right) p\left(\boldsymbol{\theta}\right)}{p\left(\boldsymbol{t}_N|\boldsymbol{X}_N\right)}$$

Having solved this inference problem (e.g. by linear regression) a prediction can be made through marginalization

$$p\left(t^{(N+1)}|\boldsymbol{t}_N, \boldsymbol{X}_N, \boldsymbol{x}^{(N+1)}\right) = \int d^H \boldsymbol{\theta} p\left(t^{(N+1)}|\boldsymbol{\theta}, \boldsymbol{x}^{(N+1)}\right) p\left(\boldsymbol{\theta}|\boldsymbol{t}_N, \boldsymbol{X}_N\right).$$

Here it is important to note that the final answer does not make any explicit reference to our parametric representation of the unknown function $y(\boldsymbol{x})$.

Assuming that we have a fixed set of basis functions and Gaussian prior distributions (with zero mean) on the weights $\boldsymbol{\theta}$ we will show that:

- The joint pdf of the observed data given the model $p(\boldsymbol{t}_N|\boldsymbol{X}_N)$, is a multivariate Gaussian with mean zero and with a covariance matrix that is determined by the basis functions.
- This implies that the conditional distribution $p(t^{(N+1)}|\boldsymbol{t}_N, \boldsymbol{X}_{N+1})$, is also a multivariate Gaussian whose mean depends linearly on $\boldsymbol{t}_N$.

**Proof.** To be added.

**The covariance matrix as the central object.** To be added.

## 0.3 Non-parametric approach: Mean and covariance functions

Similarly to how a D-dimensional Gaussian is parameterized by its mean vector and its covariance matrix, a GP is parameterized by a mean *function* and a covariance *function*. To explain this, we'll assume (without loss of generality) that the mean function is $\mu(x) = \mathbf{0}$. As for the covariance function, $C(x, x')$, it is a function that receives as input two locations $x, x'$ belonging to the input domain, i.e. $x, x' \in \mathcal{X}$, and returns the value of their co-variance.

In this way, if we have a *finite* set of input locations we can evaluate the covariance function at every pair of locations and obtain a covariance matrix $\mathbf{C}$. We write:

$$\mathbf{C} = C(\mathbf{X}, \mathbf{X}),$$

where $\mathbf{X}$ is the collection of training inputs.

We'll see below that the covariance function is what encodes our assumption about the GP. By selecting a covariance function, we are making implicit assumptions about the shape of the function we wish to encode with the GP, for example how smooth it is.

Even if the covariance function has a parametric form, combined with the GP it gives us a nonparametric model. In other words, the covariance function is specifying the general properties of the GP function we wish to encode, and not a specific parametric form for it.

**Stationary kernels.** To be added.

Below we define two very common covariance functions: The RBF (also known as Exponentiated Quadratic or Gaussian kernel) which is differentiable infinitely many times (hence, very smooth),

$$k_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left(-\frac{1}{2\ell^2}\sum_{q=1}^{Q}(x_{i,q} - x_{j,q})^2\right)$$

where $Q$ denotes the dimensionality of the input space. Its parameters are: the *lengthscale*, $\ell$ and the variance $\sigma^2$.

Furthermore, the linear kernel:

$$k_{lin}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \mathbf{x}_i^T \mathbf{x}_j$$

## 0.4 GP models for regression

To be added.

**Optimizing the GP model hyperparameters.** To be added.

## 0.5 GP emulators

To be added.