Nonparametric regression

Course of Machine Learning
Master Degree in Computer Science
University of Rome “Tor Vergata”
a.a. 2023-2024

Giorgio Gambosi

Fully bayesian regression

We remind that, in fully bayesian regression, no specific model parameters $\hat{\mathbf{w}}$ are identified, to be applied in prediction as

$$h(\mathbf{x}, \hat{\mathbf{w}}) = \hat{\mathbf{w}}^T \phi(\mathbf{x})$$

where

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_m(\mathbf{x}) \end{pmatrix}$$

Instead the distribution $p(t|\mathbf{x})$ is derived, under the assumption of gaussianity, with

$$p(t|\mathbf{x}, \Phi, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|m(\mathbf{x}), \sigma^2(\mathbf{x}))$$

and

$$m(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S} \Phi^T \mathbf{t}$$
$$\sigma^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S} \boldsymbol{\phi}(\mathbf{x})$$

where

$$\mathbf{S} = (\alpha \mathbf{I} + \beta \Phi^T \Phi)^{-1} \in \mathbb{R}^{m \times m}$$

that is,

$$\mathbf{S}^{-1} = \begin{pmatrix} \sum_{i=1}^n \beta\phi_1(\mathbf{x}_i)^2 + \alpha & \sum_{i=1}^n \beta\phi_1(\mathbf{x}_i)\phi_2(\mathbf{x}_1) & \cdots & \sum_{i=1}^n \beta\phi_1(\mathbf{x}_i)\phi_m(\mathbf{x}_1) \\ \sum_{i=1}^n \beta\phi_2(\mathbf{x}_i)\phi_2(\mathbf{x}_1) & \sum_{i=1}^n \beta\phi_2(\mathbf{x}_i)^2 + \alpha & \cdots & \sum_{i=1}^n \beta\phi_2(\mathbf{x}_i)\phi_m(\mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n \beta\phi_m(\mathbf{x}_i)\phi_1(\mathbf{x}_1) & \sum_{i=1}^n \beta\phi_m(\mathbf{x}_i)\phi_2(\mathbf{x}_1) & \cdots & \sum_{i=1}^n \beta\phi_m(\mathbf{x}_i)^2 + \alpha \end{pmatrix}$$

The prediction $h(\mathbf{x})$ can be returned here as the expectation of the predictive distribution, that is

$$h(\mathbf{x}) = m(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S} \Phi^T \mathbf{t}$$

Since

$$\Phi^T\mathbf{t} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \cdots & \phi_2(\mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ \phi_m(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n \phi_1(\mathbf{x}_i)t_i \\ \sum_{i=1}^n \phi_2(\mathbf{x}_i)t_i \\ \vdots \\ \sum_{i=1}^n \phi_m(\mathbf{x}_i)t_i \end{pmatrix} = \sum_{i=1}^n \begin{pmatrix} \phi_1(\mathbf{x}_i) \\ \phi_2(\mathbf{x}_i) \\ \vdots \\ \phi_m(\mathbf{x}_i) \end{pmatrix} t_i = \sum_{i=1}^n \boldsymbol{\phi}(\mathbf{x}_i)t_i$$

we may also write

$$h(\mathbf{x}) = \beta\boldsymbol{\phi}(\mathbf{x})^T\mathbf{S}\sum_{i=1}^n \boldsymbol{\phi}(\mathbf{x}_i)t_i = \sum_{i=1}^n \beta\boldsymbol{\phi}(\mathbf{x})^T\mathbf{S}\boldsymbol{\phi}(\mathbf{x}_i)t_i$$

We may note here that the prediction is not computed by referring to a set of parameters derived by optimization of a loss function. Instead, it can be seen as a linear combination of the target values t_i of all items in the training set, with weights dependent from the item values $\mathbf{x}_i$ (and from $\mathbf{x}$).

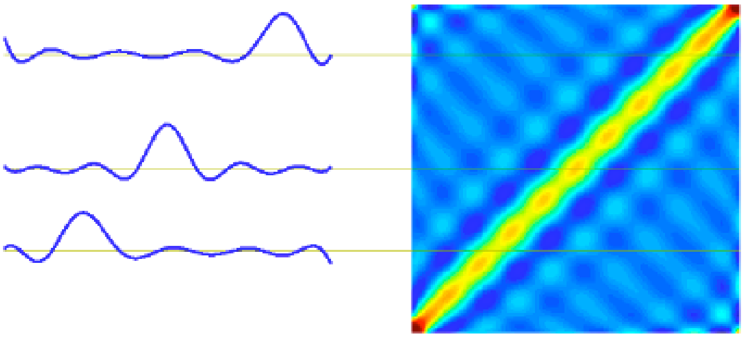
Let us denote as $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \beta\boldsymbol{\phi}(\mathbf{x}_1)^T\mathbf{S}\boldsymbol{\phi}(\mathbf{x}_2)$ the function which provides the weight associated to target value t_i, when its arguments are $\mathbf{x}_i$ and $\mathbf{x}$. Then,

$$h(\mathbf{x}) = \sum_{i=1}^n \kappa(\mathbf{x}, \mathbf{x}_i)t_i$$

The weight function $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ defined above is said equivalent kernel. Note that, in a sense, it provides a measure of how much the values of the targets associated to $\mathbf{x}_1$ and $\mathbf{x}_2$ are dependendent from each other.

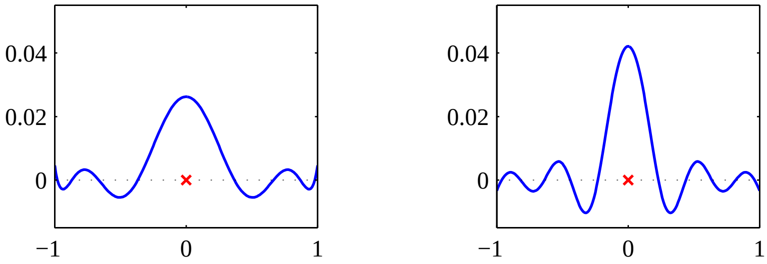
In our framework, $\kappa(\mathbf{x}, \mathbf{x}_i)$ is then a measure of how much the value of the target associated to $\mathbf{x}$, which must be approximated, is related to the target of $\mathbf{x}_i$, which is known.

In the figure below, it is shown on the right a plot on the plane (x, x_i) of a sample equivalent kernel for the case when only one feature is given, in the case when $\boldsymbol{\phi}$ is a set of gaussian base functions. On the left, a plot of the values of $\kappa(x, x_i)$ as functions of x, for three different values of x_i.



In deriving $h(x)$, the equivalent kernel tends to assign greater relevance to the target values t_i corresponding to items x_i near x.

The same localization property holds also for different base functions, as shown in the figure below, where $\kappa(0, x)$ is plotted in the case of $\boldsymbol{\phi}$ a polynomial function (left) and a gaussian function (right).



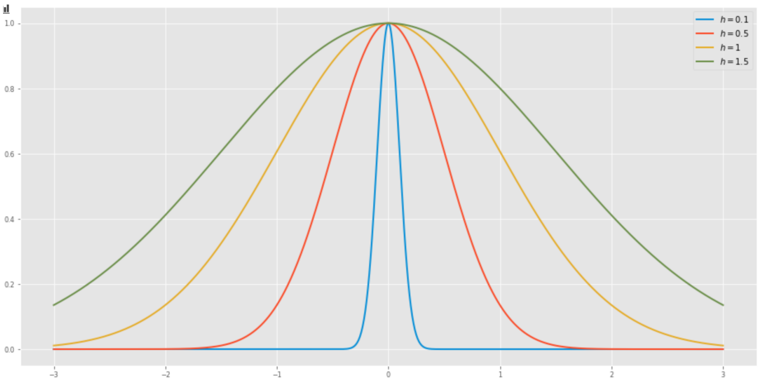
Let us finally observe that, instead of introducing base functions which eventually result into an equivalent kernel, we may follow the same approach of predicting by means of a linear combination of target values, with weights computed by a suitable localized kernel, defined on a pair of elements (that is on $\mathbb{R}^d \times \mathbb{R}^d$) and returning a real value.

Kernel regression

In kernel regression methods, the target value corresponding to any item $\mathbf{x}$ is predicted by referring to items in the training set, and in particular to the items which are closer to $\mathbf{x}$. This is controlled by referring to a predefined kernel function $\kappa_h(\mathbf{x})$, which returns non negligible values only in an interval around 0.

A possible, common kernel, is the gaussian (or RBF) kernel, plotted below in the case $d = 1$ for different values of the hyperparameter h.

$$g(\mathbf{x}) = e^{-\frac{\|\mathbf{x}\|^2}{2h^2}}$$



In order to derive the prediction function h, we remind that in regression our aim is to approximate the conditional expectation

$$E[t|\mathbf{x}] = \int t\ p(t|\mathbf{x})dt = \int t\ \frac{p(\mathbf{x},t)}{p(\mathbf{x})}dt = \frac{\displaystyle\int t\ p(\mathbf{x},t)dt}{p(\mathbf{x})} = \frac{\displaystyle\int t\ p(\mathbf{x},t)dt}{\displaystyle\int p(\mathbf{x},t)dt}$$

Assume now that the joint distribution $p(\mathbf{x}, t)$ is approximated by means of a kernel function as

$$p(\mathbf{x},t) \approx \frac{1}{n}\sum_{i=1}^{n}\kappa_h(\mathbf{x}-\mathbf{x}_i)\kappa_h(t-t_i)$$

This results into

$$h(\mathbf{x}) = \frac{\displaystyle\int t\ \frac{1}{n}\sum_{i=1}^{n}\kappa_t(\mathbf{x}-\mathbf{x}_i)\kappa_h(t-t_i)dt}{\displaystyle\int \frac{1}{n}\sum_{i=1}^{n}\kappa_h(\mathbf{x}-\mathbf{x}_i)\kappa_h(t-t_i)dt} = \frac{\sum_{i=1}^{n}\kappa_h(\mathbf{x}-\mathbf{x}_i)\displaystyle\int t\ \kappa_h(t-t_i)dt}{\sum_{i=1}^{n}\kappa_h(\mathbf{x}-\mathbf{x}_i)\displaystyle\int \kappa_h(t-t_i)dt}$$

If we assume that the kernel $\kappa(x)$ is always non negative, has mean $\displaystyle\int t\kappa_h(x)dx = 0$ and area under the curve $\displaystyle\int \kappa_h(x)dx = 1$ (which implies it is a probability density distribution), we have that $\displaystyle\int \kappa_h(t-t_i)dt = 1$ and $\displaystyle\int t\kappa_h(t-t_i)dt = t_i$, and we finally get

$$h(\mathbf{x}) = \frac{\sum_{i=1}^n \kappa_h(\mathbf{x}-\mathbf{x}_i)t_i}{\sum_{i=1}^n \kappa_h(\mathbf{x}-\mathbf{x}_i)}$$

By setting

$$w_i(\mathbf{x}) = \frac{\kappa_h(\mathbf{x}-\mathbf{x}_i)}{\sum_{j=1}^n \kappa_h(\mathbf{x}-\mathbf{x}_j)}$$

we can then write

$$h(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x})t_i$$

that is, the predicted value is computed as a normalized linear combination of all target values, weighted by applying the chosen kernel (Nadaraya-Watson).

Clearly, if base functions are applied, we get

$$h(\mathbf{x}) = \sum_{i=1}^n w_i(\boldsymbol{\phi}(\mathbf{x}))t_i$$

with

$$w_i(\mathbf{x}) = \frac{\kappa_h(\boldsymbol{\phi}(\mathbf{x})-\boldsymbol{\phi}(\mathbf{x}_i))}{\sum_{j=1}^n \kappa_h(\boldsymbol{\phi}(\mathbf{x})-\boldsymbol{\phi}(\mathbf{x}_i))}$$

Locally weighted regression

In Nadaraya-Watson model, the prediction is performed by means of a normalized weighted combination of constant values (target values in the training set).

Locally weighted regression (LOESS) improves that approach by referring to a weighted version of the sum of squared differences loss function used in regression.

If a value t has to be predicted for an item $\mathbf{x}$, a "local" version of the loss function is considered, with weight $\kappa_i(\mathbf{x})$. Assuming again base functions $\boldsymbol{\phi}$,

$$L(\mathbf{x}) = \sum_{i=1}^n \psi_i(\mathbf{x})(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2 = \sum_{i=1}^n \kappa_h(\boldsymbol{\phi}(\mathbf{x})-\boldsymbol{\phi}(\mathbf{x}_i))(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})_i - t_i)^2$$

Weights $\psi_i(\mathbf{x})$ are dependent from the "distance" between $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{x}_i)$, as measured by the kernel function

$$\psi_i(\mathbf{x}) = \kappa_h(\boldsymbol{\phi}(\mathbf{x})-\boldsymbol{\phi}(\mathbf{x}_i))$$

The minimization of this loss function

$$\hat{\mathbf{w}}(\mathbf{x}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n \psi_i(\mathbf{x})(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2$$

has solution

$$\hat{\mathbf{w}}(\mathbf{x}) = (\Phi^T\Psi(\mathbf{x})\Phi)^{-1}\Phi^T\Psi(\mathbf{x})\mathbf{t}$$

where $\Psi(\mathbf{x})$ is a diagonal $n \times n$ matrix with $\Psi(\mathbf{x})_{ii} = \kappa_i(\mathbf{x})$.

The prediction is then performed as usual, as

$$h(\mathbf{x}) = \hat{\mathbf{w}}(\mathbf{x})^T\boldsymbol{\phi}(\mathbf{x})$$

Local logistic regression

The same approach applied in the case of local regression can be applied for classification, by defining a weighted loss function to be minimized, with weights dependent from the item whose target must be predicted.

In this case, a weighted version of the cross entropy function is considered, which has to be maximized

$$L(\mathbf{x}) = \sum_{i=1}^{n} \kappa_h(\boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{\phi}(\mathbf{x})_i)(t_i \log p_i - (1 - t_i) \log(1 - p_i))$$

with $p_i = \sigma(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))$, as usual.

Gaussian processes

An alternative and equivalent way of reaching identical results to the previous ones is possible by considering inference directly in the space of functions $f : \mathbb{R}^d \mapsto \mathbb{R}$. We use a Gaussian process (GP) to describe a distribution over functions.[*]

More formally:

- A stochastic process $f(\mathbf{x})$ is a collection of (possibly infinite) random variables, $\{f(\mathbf{x}) : \mathbf{x} \in \chi\}$, the values taken by function f on domain χ. Observe that f is completely described by such values.

- A stochastic process is a Gaussian process if for any finite subset $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ of χ, the function values $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$ have joint multivariate Gaussian distribution.

In the most general case, $\chi = \mathbb{R}^d$, but simpler cases, for example with finite $|\chi|$ can be considered. Note that in this case, if $|\chi| = d$, this corresponds to stating that the joint multivariate distribution $\{f(\mathbf{x}_i) : i = 1, \dots, d\}$ is a gaussian, from which, by the properties of the gaussian distribution, it derives that the distribution of any subset of points $\{f(\mathbf{x}_i) : i \in \mathbf{I} \subset \{1, \dots, d\}\}$ is itself a gaussian.

Gaussian processes are then a generalization of joint d-dimensional multivariate gaussians which extend them to infinite d.

In order to specify the gaussian process in the general case of infinite χ, we must introduce two rules which, for any set of points $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_m)$, define the distribution $p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$ of the corresponding values.

- We already know that, by assumption, the distribution $p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$ is an m-dimensional multivariate normal distribution, which is then characterized by a mean vector $\boldsymbol{\mu}_{\mathbf{X}}$ and covariance matrix Σ_X.

- For what regards the mean, we define a function $m(\mathbf{x})$ that for each point $\mathbf{x}_i$ returns the expectation of the distribution of $f(\mathbf{x}_i)$, which is gaussian since any marginal of a gaussian distribution such as $p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$ is itself a gaussian. As a consequence, $\boldsymbol{\mu}_{\mathbf{X}} = (m(\mathbf{x}_1), \dots, m(\mathbf{x}_m))$. A possible value for $\boldsymbol{\mu}_{\mathbf{X}}$ could be just the set of target values $t_1, \dots, t_m$, thats is $m(\mathbf{x}_i) = t_i$, that is assuming that the observed value for $f(\mathbf{x}_i)$ (or its approximation) provided by t_i correspondos to the expectation of $p(f(x_i))$. However, we will see later that assuming $\boldsymbol{\mu}_{\mathbf{X}} = \mathbf{0}$ does not limit the prediction capabilities of the approach, since the effect of non zero means can be later taken into account, as a final step.

- The covariance matrix derives from the application of a predefined covariance function $\kappa : \chi \times \chi \mapsto \mathbb{R}$ which associates a real value to any pair of points in χ and, in particular, to any pair in $\mathbf{X}$, hence to all elements of $\Sigma_{\mathbf{X}}$

[*] For simplicity of notation, we refer here to the original training set points $\mathbf{x}_1, \dots, \mathbf{x}_n$ instead of using the more general notation $\boldsymbol{\phi}(\mathbf{x}_1), \dots, \boldsymbol{\phi}(\mathbf{x}_n)$ with points obtained by applying a set of base functions. All the considerations below clearly apply by substituting $\boldsymbol{\phi}(\mathbf{x})$ to $\mathbf{x}$.

The covariance function κ is assumed to be a positive definite kernel: this means that for any set of distinct points $\mathbf{x}_1, \dots, \mathbf{x}_n$ it must be

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \kappa(\mathbf{x}_i, \mathbf{x}_j) > 0$$

for any choice of the constants $c_1, \dots, c_n$ such that not all c_i are equal to 0.

Equivalently, the square Gram matrix $G_{\mathbf{X}}$ defined as

$$G_{\mathbf{X}} = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_n) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_n) \\ \cdots & \cdots & \cdots & \cdots \\ \kappa(\mathbf{x}_n, \mathbf{x}_1) & \kappa(\mathbf{x}_n, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

must have positive eigenvalues. A collection of positive definite kernels is known in the literature and can be constructed by applying suitable rules.

Thus, a Gaussian process can be interpreted as a distribution over functions whose shape (smoothness, ...) is defined by κ. If points $\mathbf{x}_i$ and $\mathbf{x}_j$ are considered to be similar (that is, $\kappa(\mathbf{x}_{,i}\, \mathbf{x}_j)$ is small) the function values at these points, $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$, can be expected to be similar too.

Recap

Reassuming, given a gaussian process $p(f) = \mathcal{GP}(m, \kappa)$, for any set of items $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, the distribution of $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$ is a gaussian

$$p(f) = p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)) = \mathcal{N}(f; \boldsymbol{\mu}_{\mathbf{X}} | \Sigma_{\mathbf{X}})$$

where

- $\boldsymbol{\mu}_{\mathbf{X}} = (m(\mathbf{x}_1), \dots, m(\mathbf{x}_n))^T$
- $\Sigma_{\mathbf{X}}$ is the Gram matrix $G_{\mathbf{X}}$ wrt $\mathbf{x}_1, \dots, \mathbf{x}_n$ of a kernel function $\kappa(\mathbf{x}, \mathbf{x}')$

For any finite subset $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ of χ, we can refer to the definition of gaussian process to obtain the distribution of $(f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$. In fact:

- it is gaussian by hypothesis
- it can be seen as the marginalization of the distribution on the infinite vector of variables defined by χ

$$p(f) = \mathcal{N}(f; \boldsymbol{\mu}_{\mathbf{X}}, \Sigma_{\mathbf{X}})$$

  where $\boldsymbol{\mu}(\mathbf{X})_i = m(\mathbf{x}_i)$ and $\Sigma_{\mathbf{X}}[i, j] = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

For any finite subset $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ of χ it is possible to sample the values of $f(\mathbf{x}_1), \dots, f(\mathbf{x}_m)$ by sampling from $\mathcal{N}(f; \boldsymbol{\mu}_{\mathbf{X}}, \Sigma_{\mathbf{X}})$

Kernels

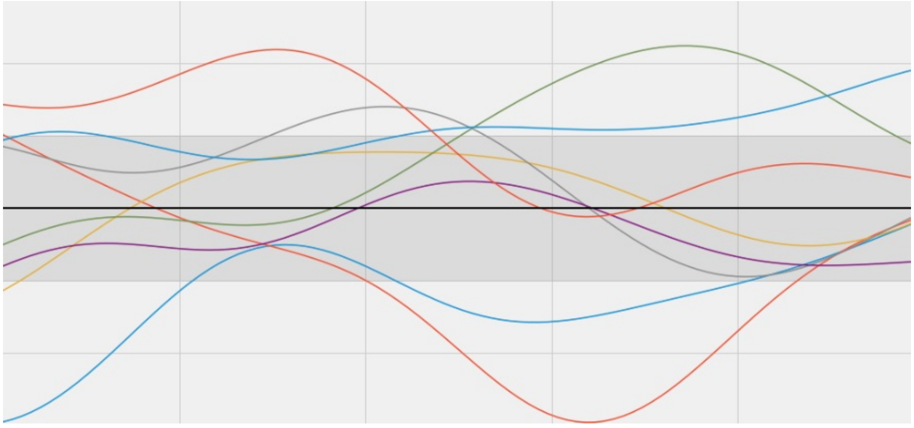
Clearly, different kernels provide different processes: one of the most applied kernel is the RBF kernel

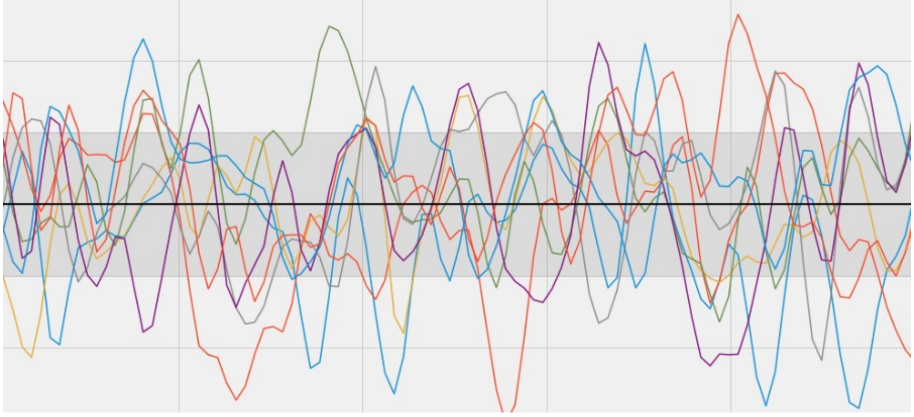
$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 e^{-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2\tau^2}}$$

which tends to assign higher covariance between $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ if $\mathbf{x}_1$ and $\mathbf{x}_2$ are nearby points.

Functions drawn from a Gaussian process with RBF kernel tend to be smooth, since values computed for nearby points tend to be similar. Smoothing is larger for larger τ.

Below, two examples of samples of functions on $\mathbb{R}$ (indeed approximated on a grid of values) are given: RBF kerenel is assumed, with larger τ in the first image and smaller τ in the second one.





Posterior distribution

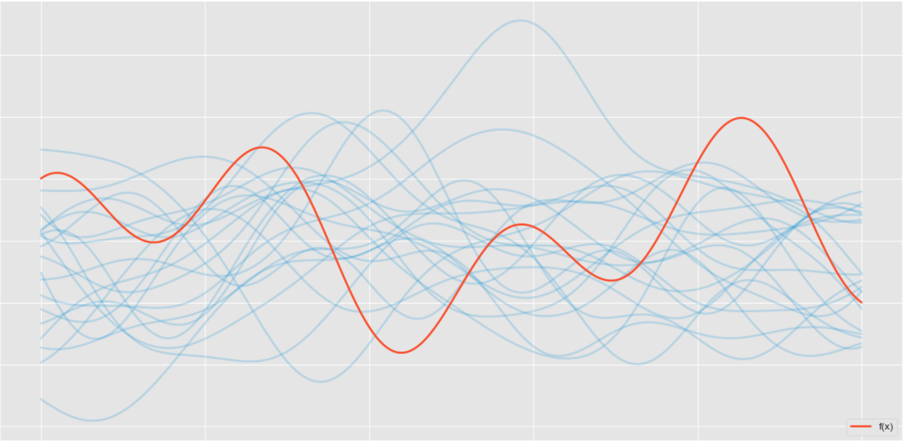
The gaussian process $\mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$ can be seen as a distribution $p(f)$ of functions, and it is independent from the actual points in the dataset. In bayesian terms, it is a prior with respect to the observation of actual values $(\mathbf{x}_i, t_i)$, where t_i is by assumption the value which is assumed is actually taken by any function sampled from $p(f)$.

This is in particular true for the set $\mathbf{X}$ of m points in the dataset. Note that here we are not taking into account the target values $\mathbf{t}$.

We have then a gaussian distribution of m-dimensional vectors, which can be interpreted as functions from $\mathbf{X}$ to $\mathbb{R}$.

$$p(f) = \mathcal{N}(f; \boldsymbol{\mu}_{\mathrm{X}}, \Sigma_{\mathrm{X}})$$

In the figure below, the red plot is the unknown function $f(x)$ to be approximated, while the thinner, blue ones are fucntions sampled by $p(f)$ ($\mathbf{X}$ is a grid of points on the x axis).



Let us now assume that each target value corresponds exactly to the value associated to point $\mathbf{x}_i$ returned by definition the unknown function f to be approximated, that is $t_i = f(\mathbf{x}_i)$. In other terms, we assume there is no noise in our observations of the unknown function f. Note that in the probabilistic model of regression this is not true, since a (gaussian) error is assumed.

By definition of gaussian process, if we now consider an additional set of points $\boldsymbol{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_r)^T$, the joint distribution of $f(\mathbf{x}_1), \dots, f(\mathbf{x}_m), f(\mathbf{z}_1), \dots, f(\mathbf{z}_r)$ is an $(m+r)$-dimensional multivariate gaussian with mean $\boldsymbol{\mu}_{(\mathrm{X},\boldsymbol{Z})} = (\boldsymbol{\mu}_{\mathrm{X}}, \boldsymbol{\mu}_{\boldsymbol{Z}})$ and covariance matrix

$$\Sigma_{(\mathrm{X},\mathrm{Z})} = \begin{pmatrix} G_{\mathrm{X}} & G_{\mathrm{Z},\mathrm{X}} \\ G^T_{\mathrm{Z},\mathrm{X}} & G_{\mathrm{Z}} \end{pmatrix}$$

where

$$G_{\mathrm{Z},\mathrm{X}} = \begin{pmatrix} \kappa(\mathbf{z}_1,\mathbf{x}_1) & \kappa(\mathbf{z}_1,\mathbf{x}_2) & \cdots & \kappa(\mathbf{z}_1,\mathbf{x}_m) \\ \kappa(\mathbf{z}_2,\mathbf{x}_1) & \kappa(\mathbf{z}_2,\mathbf{x}_2) & \cdots & \kappa(\mathbf{z}_2,\mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{z}_r,\mathbf{x}_1) & \kappa(\mathbf{z}_m,\mathbf{x}_2) & \cdots & \kappa(\mathbf{z}_r,\mathbf{x}_m) \end{pmatrix}$$

We wish to derive the predictive distribution of $f(\mathbf{z}_1), \dots, f(\mathbf{z}_r)$ given $\mathbf{z}_1, \dots, \mathbf{z}_r, \mathbf{x}_1, \dots, \mathbf{x}_m$, and $t_1, \dots, t_m$, which by the no noise assumption is equal to $f(\mathbf{x}_1), \dots, f(\mathbf{x}_m)$. That is, we wish to derive the conditional distribution $p(f(\boldsymbol{Z})|\boldsymbol{Z}, \mathrm{X}, f(\mathrm{X}))$. In order to do this, let us first remind some useful properties of multivariate gaussian distributions.

Recap: some properties of Gaussian distributions

Let $\mathrm{x} = (x_1, \dots, x_n)^T$ be a random vector with gaussian distribution $p(\mathrm{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and let $\mathrm{x} = (\mathrm{x}_A, \mathrm{x}_B)$ be a partition of the components x such that:

- $\mathrm{x}_A = (x_1, \dots, x_r)^T$
- $\mathrm{x}_B = (x_{r+1}, \dots, x_n)^T$

Then, the marginal distributions $p(\mathrm{x}_A)$ and $p(\mathrm{x}_B)$ are both gaussian with means $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B$ and covariance matrices Σ_A, Σ_B which can be derived from $\boldsymbol{\mu}, \Sigma$ by observing that

$$\boldsymbol{\mu} = (\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)^T \qquad\qquad \Sigma = \begin{pmatrix} \Sigma_A & \Sigma_{AB} \\ \Sigma^T_{AB} & \Sigma_B \end{pmatrix}$$

Clearly, $\boldsymbol{\mu}_A \in \mathbb{R}^r$, $\boldsymbol{\mu}_B \in \mathbb{R}^{n-r}$, $\Sigma_A \in \mathbb{R}^{r\times r}$, $\Sigma_B \in \mathbb{R}^{(n-r)\times(n-r)}$,

In the same situation, the conditional distributions $p(\mathrm{x}_A|\mathrm{x}_B)$ and $p(\mathrm{x}_B|\mathrm{x}_A)$ are also gaussian with means

$$\boldsymbol{\mu}_{A|B} = \boldsymbol{\mu}_A + \Sigma_{AB}\Sigma_B^{-1}(\mathrm{x}_B - \boldsymbol{\mu}_B)$$
$$\boldsymbol{\mu}_{B|A} = \boldsymbol{\mu}_B + \Sigma_{BA}\Sigma_A^{-1}(\mathrm{x}_A - \boldsymbol{\mu}_A)$$

and covariance matrices

$$\Sigma_{A|B} = \Sigma_A - \Sigma_{AB}\Sigma_B^{-1}\Sigma_{BA}$$
$$\Sigma_{B|A} = \Sigma_B - \Sigma_{BA}\Sigma_A^{-1}\Sigma_{AB}$$

From these properties, by setting $\mathbf{x}_A = f(\mathrm{X})$ and $\mathbf{x}_B = f(\boldsymbol{Z})$, it results that

$$p(f(\boldsymbol{Z})|\boldsymbol{Z}, \mathrm{X}, f(\mathrm{X})) = p(f(\mathrm{z}_1), \ldots, f(\mathrm{z}_r)|\mathrm{z}_1, \ldots, \mathrm{z}_r, \mathrm{x}_1, \ldots, \mathrm{x}_m, f(\mathrm{x}_1), \ldots, f(\mathrm{x}_m))$$

is an r-dimensional gaussian distribution itself with mean and covariance defined as

$$\boldsymbol{\mu}_{pr} = \boldsymbol{\mu}_{\mathrm{Z}} + G_{\mathrm{Z,X}}G_{\mathrm{X}}^{-1}(\mathbf{t} - \boldsymbol{\mu}_{\mathrm{X}})$$
$$\Sigma_{pr} = G_{\mathrm{Z}} - G_{\mathrm{Z,X}}G_{\mathrm{X}}^{-1}G_{\mathrm{Z,X}}^T$$

Observe that even under the simplifying assumption that $(Vmu_{\mathrm{X}}, \boldsymbol{\mu}_{\boldsymbol{Z}}) = \mathbf{0}$, that is that $m(\mathbf{x})$ is assumed 0 for all $\mathbf{x}$ (and $\mathbf{z}$), the mean of the predictive distribution may result to be non zero. In fact, in such a case, it would be

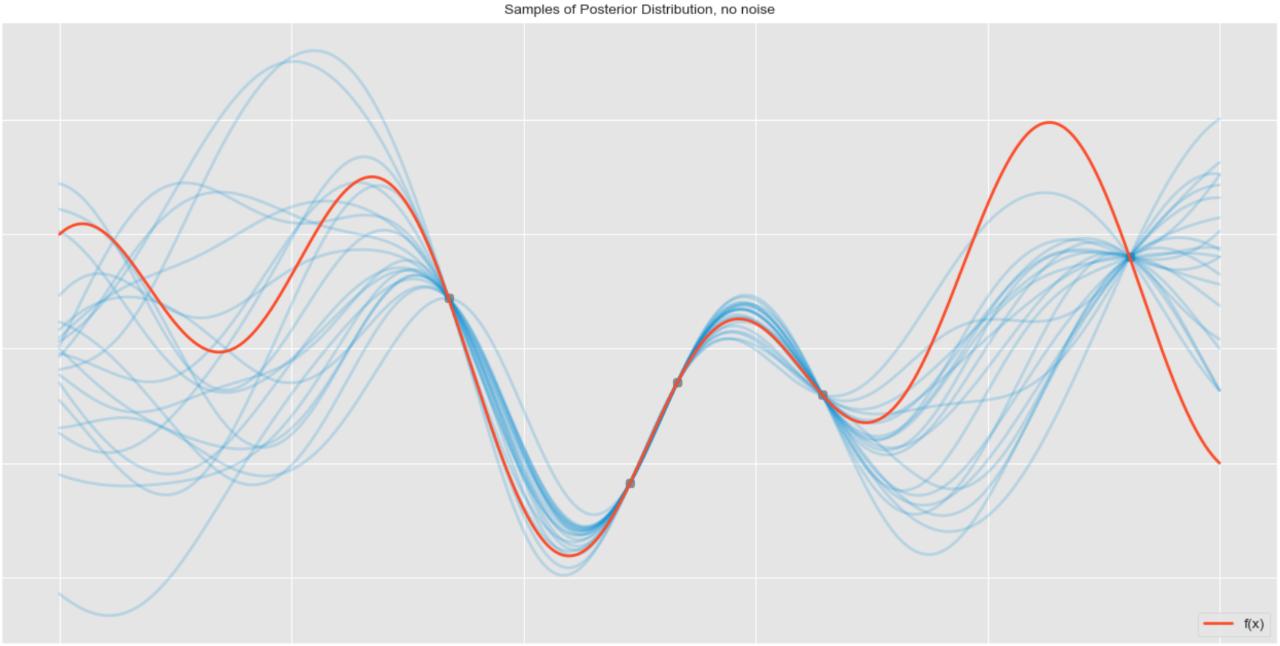
$$\boldsymbol{\mu}_{pr} = G_{\mathrm{Z,X}}G_{\mathrm{X}}^{-1}\mathbf{t}$$

However, by the first equation above, even in the general case of any definition of $m(\mathbf{x})$, we may assume that $m(\mathbf{x}) = 0$, obtaining $\boldsymbol{\mu}_{pr} = G_{\mathrm{Z,X}}G_{\mathrm{X}}^{-1}\mathbf{t}$, and next modify such value as

$$\boldsymbol{\mu}_{pr} = \boldsymbol{\mu}_{pr} + \boldsymbol{\mu}_{\boldsymbol{Z}} - G_{\mathrm{Z,X}}G_{\mathrm{X}}^{-1}\boldsymbol{\mu}_{\mathrm{X}}$$

to take into account the assumed non zero expectations. This shows that we could have indeed considered the case $m(\mathbf{x}) = 0$ in the above considerations without loss of generality.

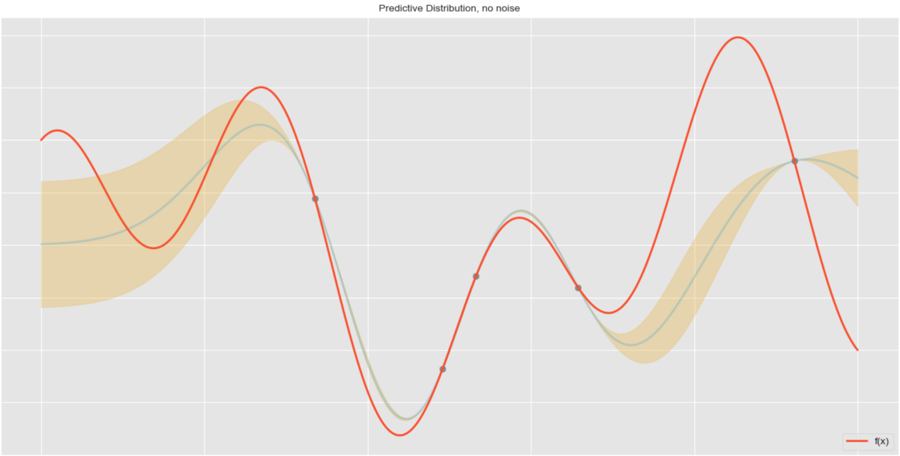
Sampling several functions from such the predictive distribution results in the following situation: again, the red plot is the unknown functio f whose values at 5 points are now known, while the blue plot are samples from the posterior distribution $\mathcal{N}(x|\boldsymbol{\mu}_{pr}, \Sigma_{pr})$. Observe that all such functions have the same values of f at the 5 points.



The same considerations apply, in particular, for the prediction of a single test point $\mathbf{z}$ given the training set $\mathbf{X}, \mathbf{t}$. According to what shown above, the predictive distribution of $f(\mathbf{x})$ is a gaussian distribution with mean and variance

$$\begin{aligned} \mu_{pr} &= G_{\mathbf{z},\mathbf{X}} G_{\mathbf{X}}^{-1}(\mathbf{t} - \boldsymbol{\mu}_{\mathbf{X}}) \\ \sigma_{pr}^2 &= \kappa(\mathbf{z}, \mathbf{z}) - G_{\mathbf{z},\mathbf{X}} G_{\mathbf{X}}^{-1} G_{\mathbf{z},\mathbf{X}}^T \end{aligned}$$

In the figure below, the mean value of the predictive distribution of $f(x)$ for each point x, given the 5 points shown on the red plot, is shown as a blue plot, with the corresponding variance reported by the yellow interval around such plot.



As already observed, in this case an interpolation of the given values has been performed, namely $f(\mathbf{x}_i) = t_i$ for all possible functions, sampled from $p(f|\mathbf{X}, \mathbf{t})$.

It results, in fact, for all $\mathbf{x}_i \in \mathbf{X}$,

$$\begin{aligned} \mu(f(\mathbf{x}_i)|\mathbf{X}, \mathbf{t}) &= t_i \\ \sigma^2 &= 0 \end{aligned}$$

Gaussian process regression: gaussian noise

If we make the more realistic hypothesis that each target value t_i only provides a noisy observation of $f(\mathbf{x}_i)$, we may behave as in the definition of the probabilistic model for linear regression: in particular, we may make the hypothesis of a gaussian noise, hence that $p(t_i|f, \mathbf{x}_i) = \mathcal{N}(f(\mathbf{x}_i), \sigma_f^2)$, while earlier we assumed $t_i = f(\mathbf{x}_i)$.

Then the value t_i observed for variable $\mathbf{x}_i$ differs from the one obtained as $f(\mathbf{x}_i)$ by a gaussian and independent noise

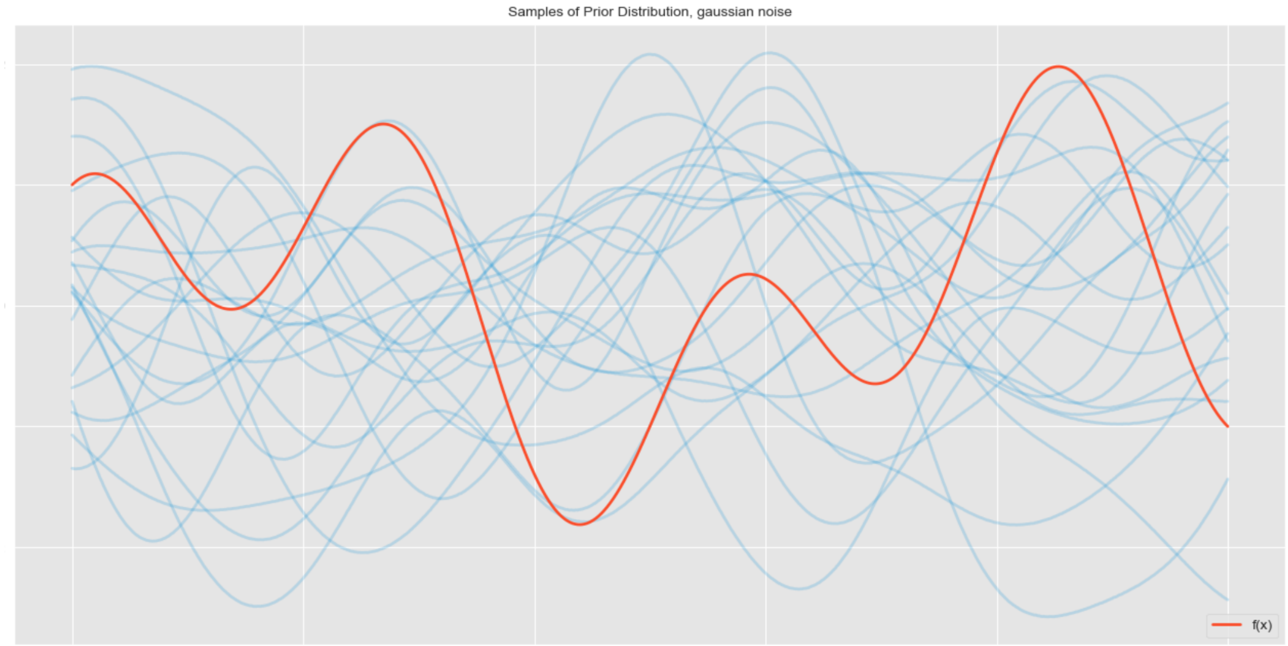
$$t_i = f(\mathbf{x}_i) + \varepsilon \qquad\qquad p(\varepsilon) = \mathcal{N}(\varepsilon; 0, \sigma_f^2)$$

Under these assumptions, for the prior distribution on the noisy observations we have that the variance of $f(\mathbf{x}_i)$ is increased, with respect to the previous case, by the uncertainty derived by the noise, whcih has variance σ_f^2. As a consequence, we have that:

$$\begin{aligned} \Sigma_{\mathbf{X}}[i, j] &= \kappa(\mathbf{x}_i, \mathbf{x}_j) && \text{if } i \neq j \\ \Sigma_{\mathbf{X}}[i, i] &= \kappa(\mathbf{x}_i, \mathbf{x}_j) + \sigma_f^2 \end{aligned}$$

As a consequence, the covariance matrix Σ_{X} results

$$\Sigma(\mathrm{X}) = G_{\mathrm{X}} + \sigma_f^2 \mathbf{I}$$



Gaussian process regression: gaussian noise

Let us now assume that a training set $\mathbf{X}, \mathbf{t}$ is available such that the target values in the training set correspond approximately to the function value $t_i = f(\mathbf{x}_i) + \varepsilon$.

In this case, for any new set of points $\mathbf{Z}$, the joint distribution of $(f(\mathbf{X}), f(\boldsymbol{Z}))$ is a multivariate gaussian distribution with mean $\boldsymbol{\mu}_{(\mathrm{X},\boldsymbol{Z})} = (\boldsymbol{\mu}_{\mathrm{X}}, \boldsymbol{\mu}_{\boldsymbol{Z}})$ and covariance matrix

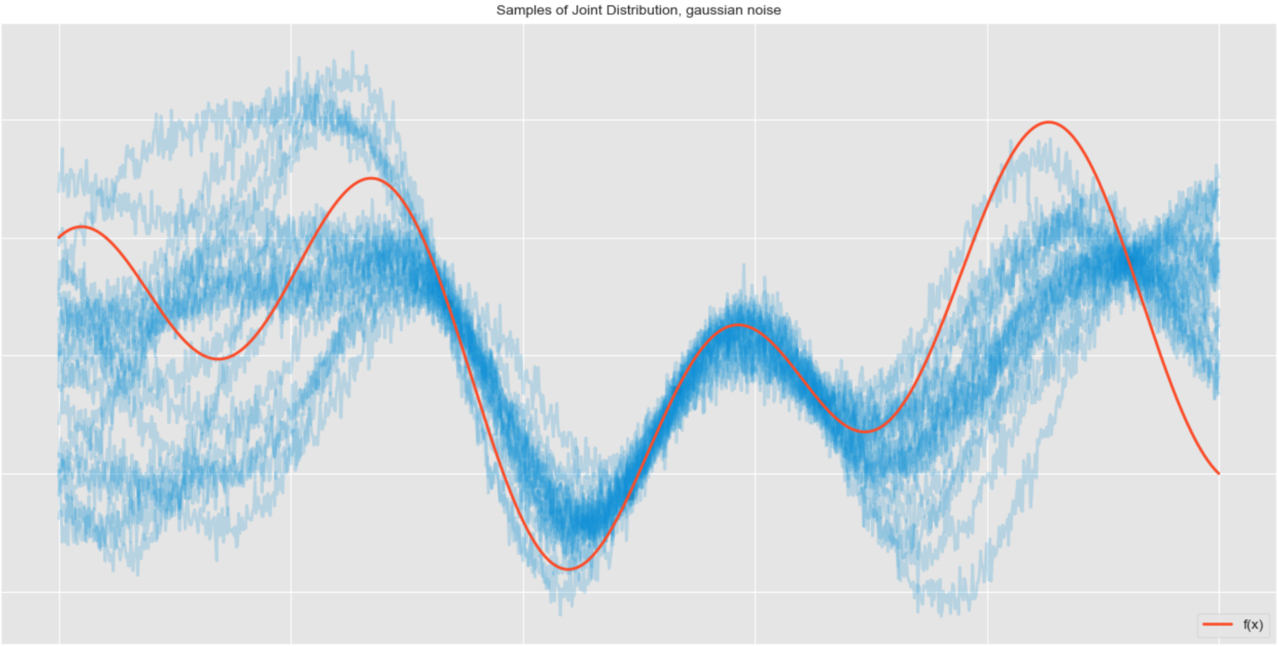
$$\hat{\Sigma}_{(\mathrm{X},\mathrm{Z})} = \begin{pmatrix} \hat{\Sigma}_{\mathrm{X}} & G_{\mathrm{Z},\mathrm{X}} \\ G_{\mathrm{Z},\mathrm{X}}^T & G_{\mathrm{Z}} \end{pmatrix}$$

where

$$\hat{\Sigma}_{\mathrm{X}} = G_{\mathrm{X}} + \sigma_f^2 \mathbf{I} = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) + \sigma_f^2 & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) + \sigma_f^2 & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \kappa(\mathbf{x}_m, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) + \sigma_f^2 \end{pmatrix}$$

The predictive distribution of $f(\mathbf{z}_1), \ldots, f(\mathbf{z}_r)$ given $\mathbf{z}_1, \ldots, \mathbf{z}_r$, $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and $t_1, \ldots, t_m$ can be again derived by the gaussian distribution properties, and, by the same considerations, turns out again to be a gaussian distribution with mean and covariance defined as

$$\hat{\boldsymbol{\mu}}_{pr} = \boldsymbol{\mu}_{\mathrm{Z}} + G_{\mathrm{Z},\mathrm{X}} \hat{\Sigma}(\mathrm{X})^{-1} (\mathbf{t} - \boldsymbol{\mu}_{\mathrm{X}})$$
$$\hat{\Sigma}_{pr} = G_{\mathrm{Z}} - G_{\mathrm{Z},\mathrm{X}} \hat{\Sigma}(\mathrm{X})^{-1} G_{\mathrm{Z},\mathrm{X}}^T$$

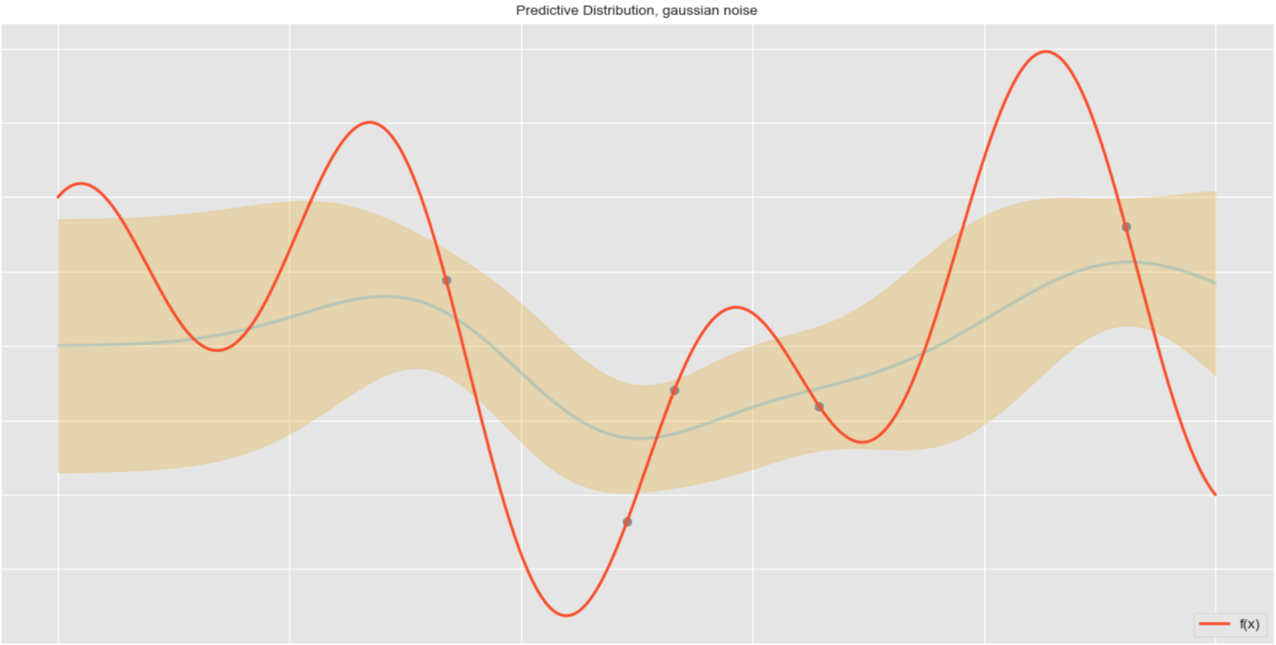


Again, if we assume zero mean in the prior distribution it results

$$\hat{\boldsymbol{\mu}}_{pr} = G_{Z,X}\hat{\Sigma}_X^{-1}\mathbf{t}$$

In particular, for a single test point $\mathbf{z}$, we have now that the corresponding predictive distribution is again a gaussian with

$$\mu_{pr} = m(\mathbf{x}) + G_{\mathbf{z},X}\hat{\Sigma}_X^{-1}(\mathbf{t} - \boldsymbol{\mu}_X)$$
$$\sigma_{pr}^2 = \kappa_p(\mathbf{z},\mathbf{z}) - G_{\mathbf{z},X}\hat{\Sigma}_X^{-1}G_{\mathbf{z},X}^T$$



Estimating kernel parameters

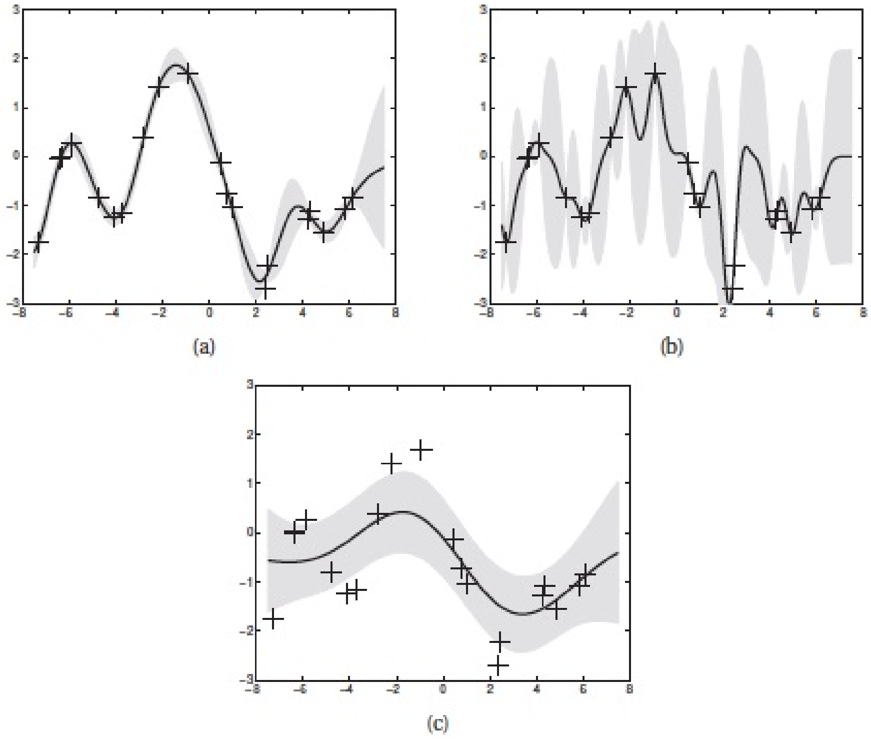
The predictive performance of gaussian processes depends exclusively on the suitability of the chosen kernel.

Let us consider the case of an RBF kernel. Then,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 e^{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)} + \sigma_y^2 \delta_{ij}$$

M can be defined in several ways: the simplest one is $\mathbf{M} = l^{-2}\mathbf{I}$.

Even in this simple case, varying the values of σ_f, σ_y, l returns quite different results.



(figure from K.Murphy "Machine learning: a probabilistic perspective" p. 519, with (l, σ_f, σ_y) equal to $(1, 1, 0.1)$, $(0.3, 1.08, 0.00005)$, $(3.0, 1.16, 0.89)$)