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

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

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

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

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

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

and

$$\Phi = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \phi_2(\mathbf{x}_n) & \cdots & \phi_m(\mathbf{x}_n) \end{pmatrix} \in \mathbb{R}^{n \times m}$$

Equivalent kernel

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

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

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

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

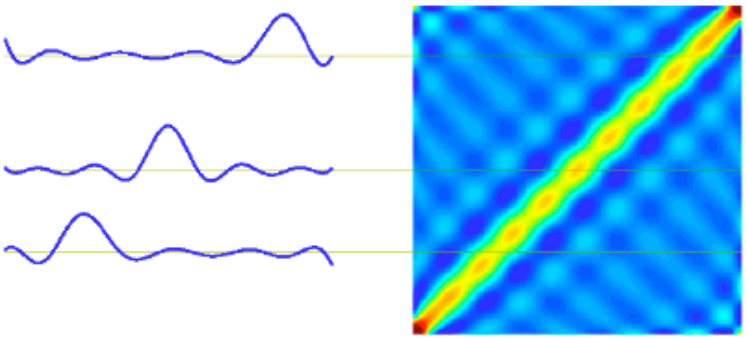
The prediction is not computed by referring to a set of parameters derived by optimization of a loss function. Instead, it can be seen as a linear combination of the target values t_i of all items in the training set, with weights dependent from the item values $\mathbf{x}_i$ (and from $\mathbf{x}$)

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

The weight function $\kappa(\mathbf{x}, \mathbf{x}') = \beta\boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}\boldsymbol{\phi}(\mathbf{x}')$ is said **equivalent kernel**

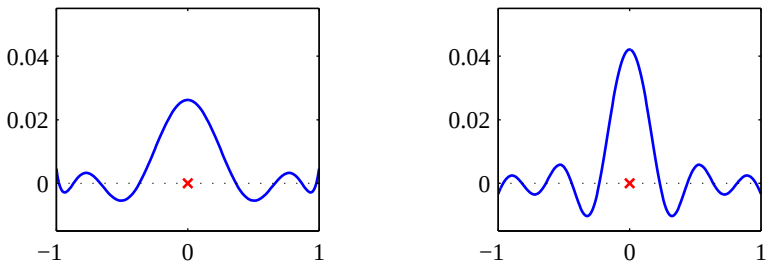
Right: plot on the plane (x, x_i) of a sample equivalent kernel, in the case of gaussian basis functions.

Left: plot as a function of x_i for three different values of x



In deriving y, 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.



Left, $\kappa(0, x')$ in the case of polynomial basis functions.
Right, $\kappa(0, x')$ in the case of gaussian basis functions.

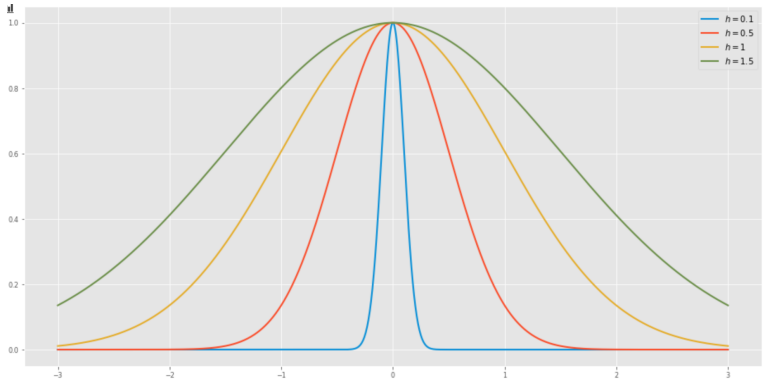
Idea: instead of introducing base functions which results into a kernel, we may define a localized kernel directly and use it to make predictions

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 **kernel** function $\kappa_h(\mathbf{x})$, which is non zero only in an interval around 0
- h is the **bandwidth** of the kernel, which controls the width of $\kappa_h(\mathbf{x})$

A possible, common kernel, is the gaussian (or RBF) kernel

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



In regression, we are interested in estimating the conditional expectation

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

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

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

If we assume that the kernel $\kappa(x)$ is a probability distribution with 0 mean, it results $\int \kappa_h(t - t_i)dt = 1$ and $\int t\kappa_h(t - t_i)dt = t_i$, we get

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

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 write

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

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

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 $\kappa_i(\mathbf{x})$.

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

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

$$\kappa_i(\mathbf{x}) = \kappa_h(\mathbf{x} - \mathbf{x}_i)$$

The minimization of this loss function

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

has solution

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

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

$$y = \hat{\mathbf{w}}(\mathbf{x})^T\bar{\mathbf{x}}$$

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

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

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

The loss function minimization can be performed, for example, by applying a suitable modification of the IRLS algorithm for logistic regression

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 $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of χ, the function values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ have joint multivariate Gaussian distribution

In the most general case, $\chi \equiv \mathbb{R}^d$, but simpler cases, for example with finite χ can be considered: Gaussian processes are a generalization of multivariate gaussians on random vectors which extend multivariate gaussians to infinite-sized collections of real-valued variables. In the case of a finite domain χ, a gaussian process is reduced to a multivariate gaussian on $\mathbb{R}^{|\chi|}$.

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 $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, define the distribution $p(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ of the corresponding values.

- We already know that, by assumption, the distribution $p(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ is a multivariate normal distribution, characterized then by a **mean vector** $\boldsymbol{\mu}_\mathbf{X}$ and **covariance matrix** Σ_X*
- We define a mean function $m(\mathbf{x})$ such that $m(\mathbf{x}) = E[f(\mathbf{x}]$, hence $\boldsymbol{\mu}_\mathbf{X} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_m)$. Usually, it is possible to safely assume $m(\mathbf{x}) = 0$
- 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 $\Sigma_\mathbf{X}$

The covariance function κ is assumed to be a **positive definite kernel**: this means that for any set of distinct points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ 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 $c_1, \ldots, c_n$ such that not all c_i are equal to 0.

*Observe that expectations are taken with respect to the random function f.

Equivalently, the square Gram matrix G_X defined as

$$G_X = \begin{pmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_n) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_n) \\ \cdots & \cdots & \cdots & \cdots \\ \kappa(x_n, x_1) & \kappa(x_n, x_2) & \cdots & \kappa(x_n, x_n) \end{pmatrix}$$

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 is a distribution over functions whose shape (smoothness, ...) is defined by κ. If points x_i and x_j are considered to be similar ($\kappa(x,_i x_j)$ is small) the function values at these points, $f(x_i)$ and $f(x_j)$, can be expected to be similar too.

We may then define the Gaussian process as

$$f(x) \sim \mathcal{GP}(m(x), \kappa(x, x'))$$

Recap

Reassuming, given a gaussian process $p(f) = \mathcal{GP}(m, \kappa)$, for any set of items $X = (x_1, \ldots, x_n)$, the distribution of $f(x_1), \ldots, f(x_n)$ is a gaussian

$$p(f) = p(f(x_1), \ldots, f(x_n) = \mathcal{N}(f; \mu_X | \Sigma_X)$$

where

- $\mu_X = (m(x_1), \ldots, m(x_n))^T$
- Σ_X is the Gram matrix G_X wrt $x_1, \ldots, x_n$ of a kernel function $\kappa(x, x')$

For any finite subset $X = (x_1, \ldots, x_m)$ of χ, we can refer to the definition of gaussian process to obtain the distribution of $(f(x_1), \ldots, f(x_m))$. 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

$$p(f) = \mathcal{N}(f; \mu_X, \Sigma_X)$$

where $\mu(X)_i = m(x_i)$ and $\Sigma_X[i, j] = \kappa(x_i, x_j)$.

For any finite subset $X = (x_1, \ldots, x_n)$ of χ it is possible to sample from $p(f)$ the values of $f(x_1), \ldots, f(x_m)$ by gaussian sampling from $\mathcal{N}(f; \mu_X, \Sigma_X)$

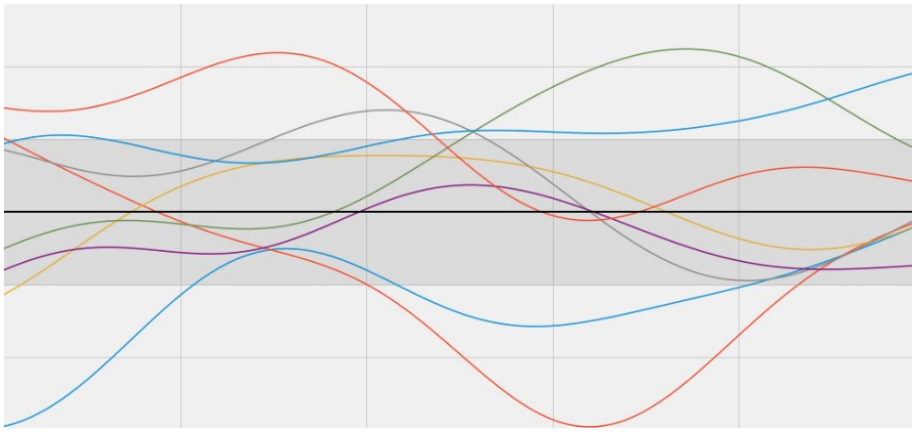
Kernels

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

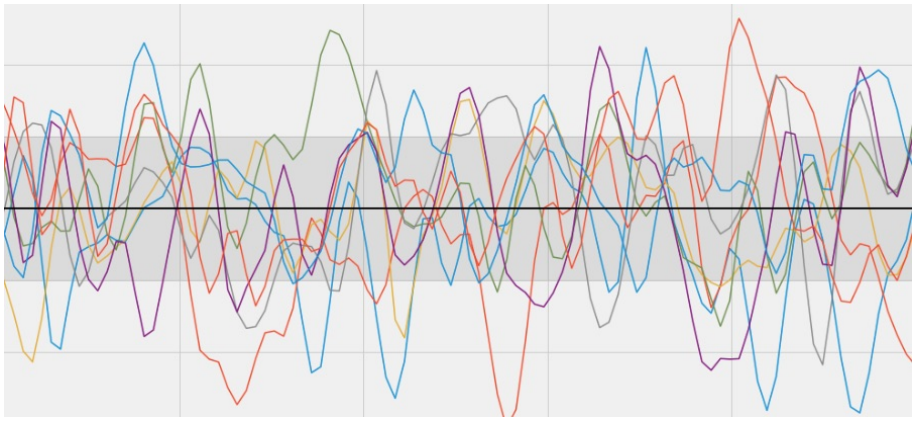
$$\kappa(x_1, x_2) = \sigma^2 e^{-\frac{||x_1 - x_2||^2}{2\tau^2}}$$

which tends to assign higher covariance between $f(x_1)$ and $f(x_2)$ if x_1 and 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.



Samples of functions from $p(f)$. RBF kernel, smaller τ and smoothing

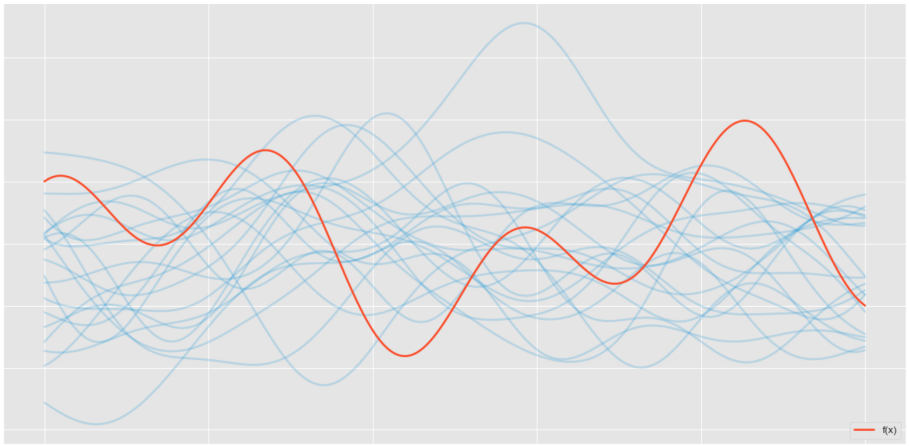


Posterior distribution

The gaussian process $\mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$ can be seen as a prior distribution of functions (prior with respect to the observation of some values $(\mathbf{x}, f(\mathbf{x}))$, i.e. of the values actually taken by the function at some points).

By the considerations above, this results in a prior distribution of function values for any finite subset of points:

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



Let us now assume that, given $\mathbf{X}$ the corresponding values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)$ are known: that is, we assume that a training set $\mathbf{X}, \mathbf{t}$ is available, and that the target values $t_1, \ldots, t_m$ correspond exactly to the values of the unknown regression function at the corresponding items, 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 $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_r)^T$, the joint distribution of $f(\mathbf{x}_1), \dots, f(\mathbf{x}_m), f(\mathbf{z}_1), \dots, f(\mathbf{z}_r)$ is a multivariate gaussian with mean $\boldsymbol{\mu}_{(\mathrm{X,Z})} = (\boldsymbol{\mu}_\mathrm{X}, \boldsymbol{\mu}_\mathrm{Z})$ and covariance matrix

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

where

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

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 $\mathbf{t} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$, that is the conditional distribution $p(f(\mathbf{Z})|\mathbf{Z}, \mathbf{X}, \mathbf{t})$. In order to do that, let us first remind some useful properties of multivariate gaussian distributions.

Recap: some properties of Gaussian distribution

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

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

Then, the marginal distributions $p(\mathbf{x}_A)$ and $p(\mathbf{x}_B)$ 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

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

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 $p(\mathbf{x}_A|\mathbf{x}_B)$ and $p(\mathbf{x}_B|\mathbf{x}_A)$ are also gaussian with means

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

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 $\mathbf{x}_A = f(\mathbf{X})$ and $\mathbf{x}_B = f(\boldsymbol{Z})$, it results that

$$p(f(\mathbf{z}_1), \dots, f(\mathbf{z}_r)|\mathbf{z}_1, \dots, \mathbf{z}_r, \mathbf{x}_1, \dots, \mathbf{x}_m, t_1, \dots, t_m)$$

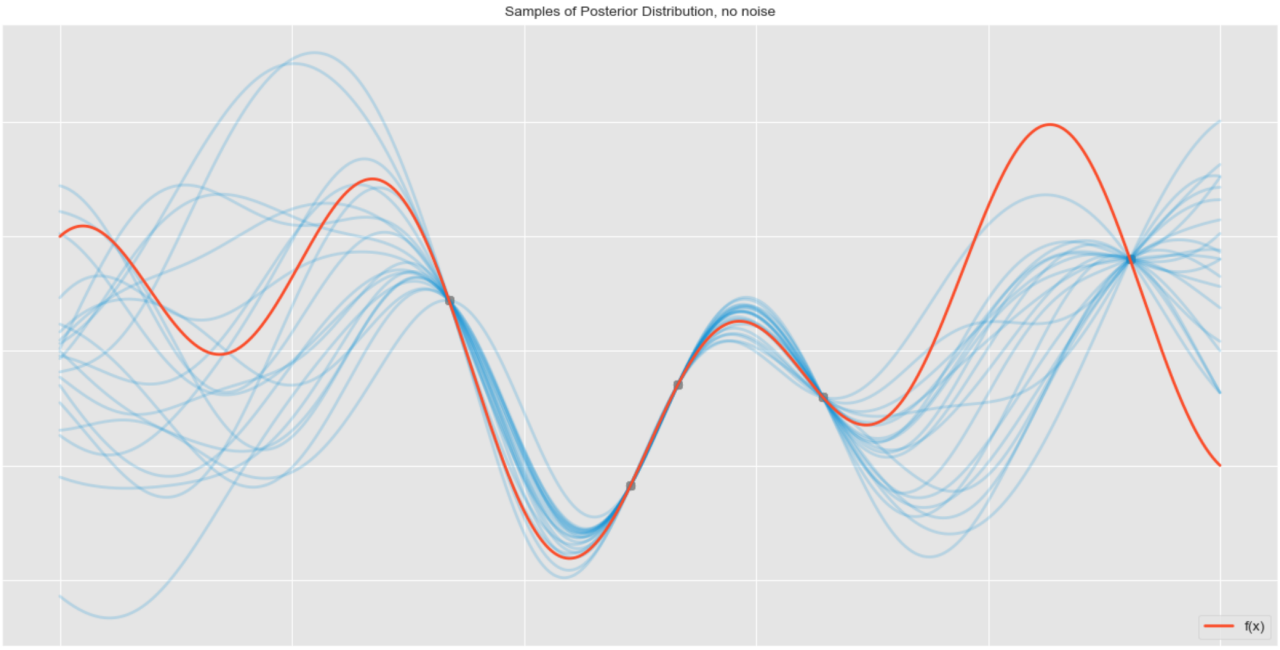
is an r-dimensional gaussian distribution itself with mean and covariance defined as

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

Observe that even under the assumption that $m(\mathbf{x}) = 0$ in the gaussian process definition the mean of the predictive distribution may result to be non zero. In fact, in such a case, it would be

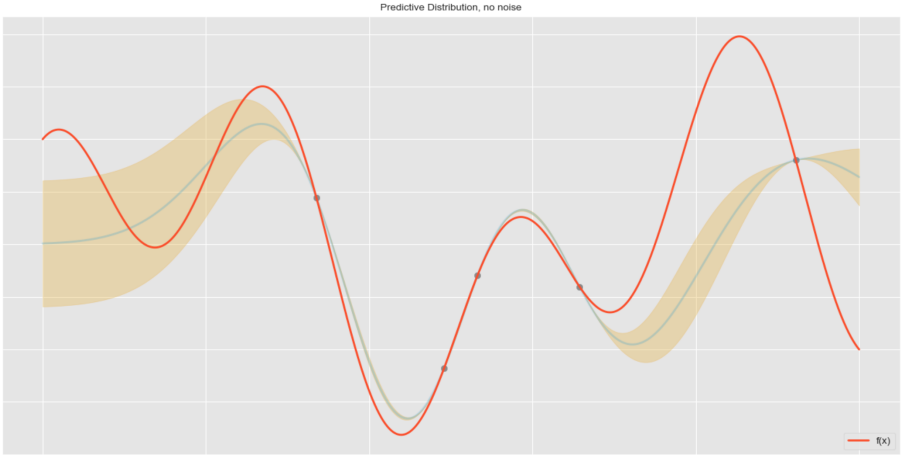
$$\boldsymbol{\mu}_{pr} = G_{\mathrm{Z,X}} G_{\mathrm{X}}^{-1} \mathbf{t}$$

Sampling several functions from such the predictive distribution results in the following situation



In particular, for the prediction of a single test point $\mathbf{z}$, the predictive distribution of $f(\mathbf{x})$ is a gaussian with mean and variance

$$\mu_{pr} = G_{\mathrm{z,X}} G_{\mathrm{X}}^{-1} \mathbf{t}$$
$$\sigma_{pr}^2 = \kappa(\mathbf{z}, \mathbf{z}) - G_{\mathrm{z,X}} G_{\mathrm{X}}^{-1} G_{\mathrm{z,X}}^{T}$$



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

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

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

Gaussian process regression: gaussian noise

If we make the more realistic hypothesis that 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, that is assume the gaussianity of noise, hence that $p(t_i|f, \mathbf{x}_i) = \mathcal{N}(f(\mathbf{x}_i), \sigma_f^2)$

That is, 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

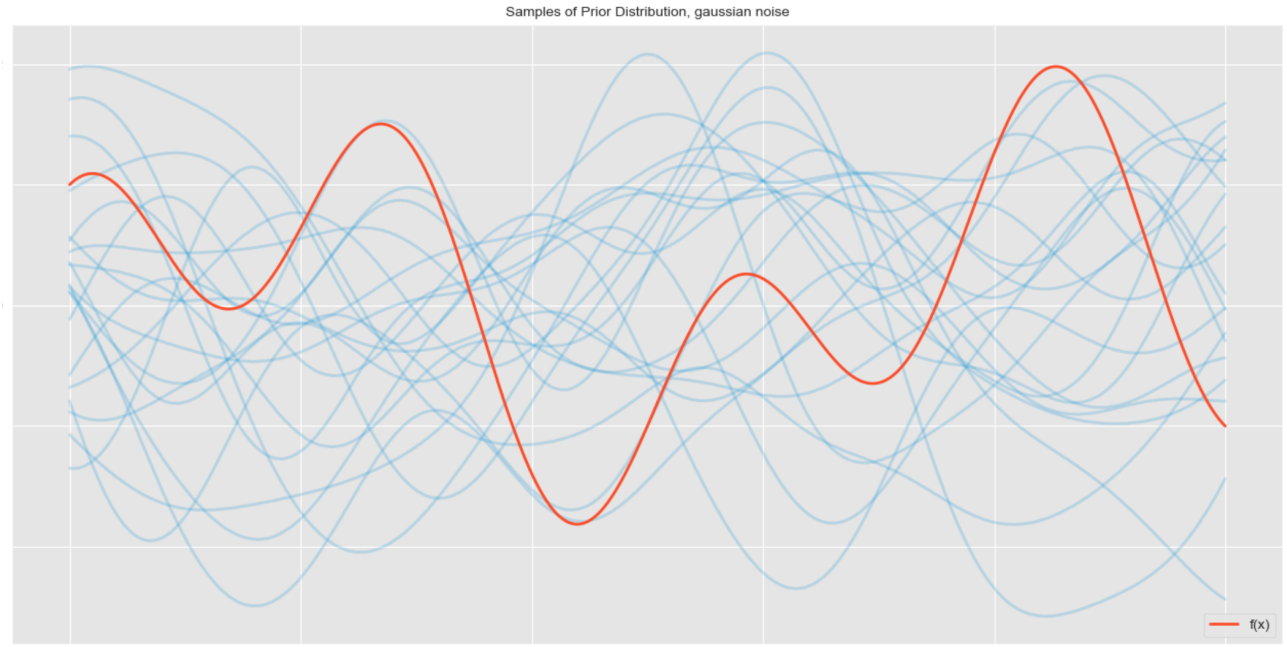
$$t_i = f(\mathbf{x}_i) + \varepsilon \qquad\qquad p(\varepsilon) = \mathcal{N}(\varepsilon; 0, \sigma_f^2)$$

Under these assumptions, for the prior distribution on the noisy observations we have

$$\text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = \kappa(\mathbf{x}_i, \mathbf{x}_j) + \sigma_f^2 \delta_{ij}$$

where δ_{ij} is the Kronecker delta which is one iff $i = j$ and zero otherwise. As a consequence, the covariance matrix $\Sigma_\mathbf{X}$ results

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



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 $(\mathbf{t}, f(\mathbf{Z}))$ is a multivariate gaussian distribution is a multivariate gaussian with mean $\boldsymbol{\mu}_{(\mathrm{X},\mathrm{Z})} = (\boldsymbol{\mu}_\mathrm{X}, \boldsymbol{\mu}_\mathrm{Z})$ and covariance matrix

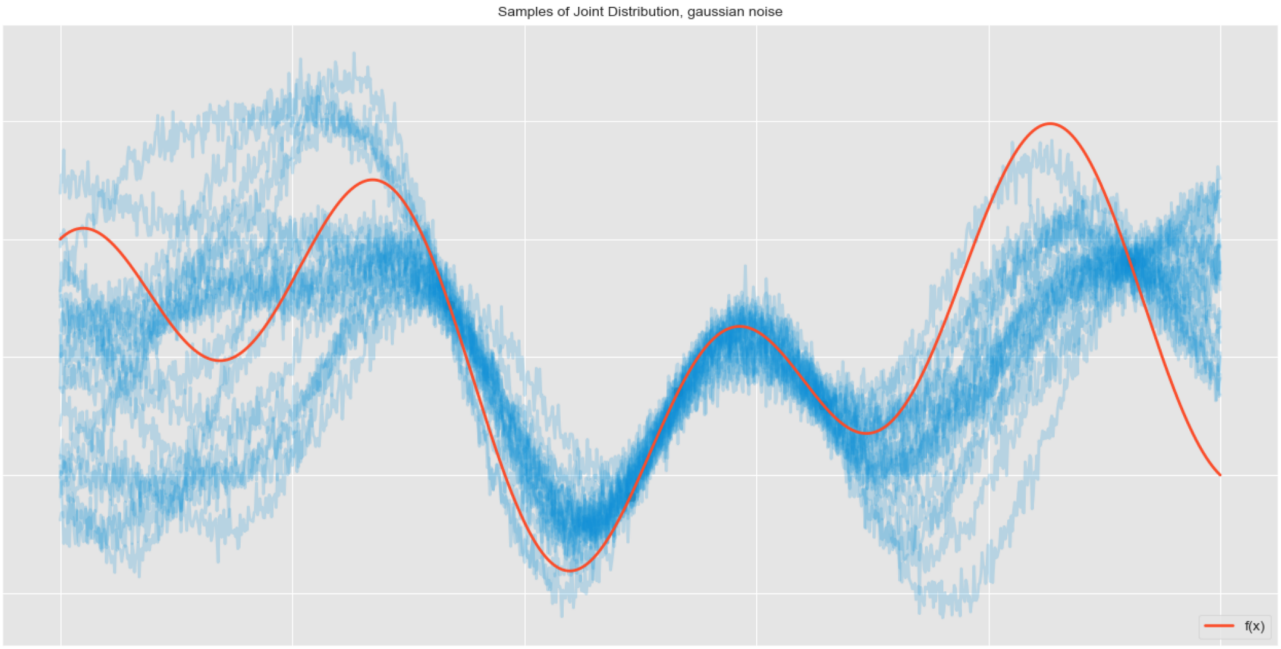
$$\hat{\Sigma}_{(\mathrm{X},\mathrm{Z})} = \begin{pmatrix} \hat{\Sigma}_\mathrm{X} & G_{\mathrm{Z},\mathrm{X}} \\ G_{\mathrm{Z},\mathrm{X}}^T & G_\mathrm{Z} \end{pmatrix}$$

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

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

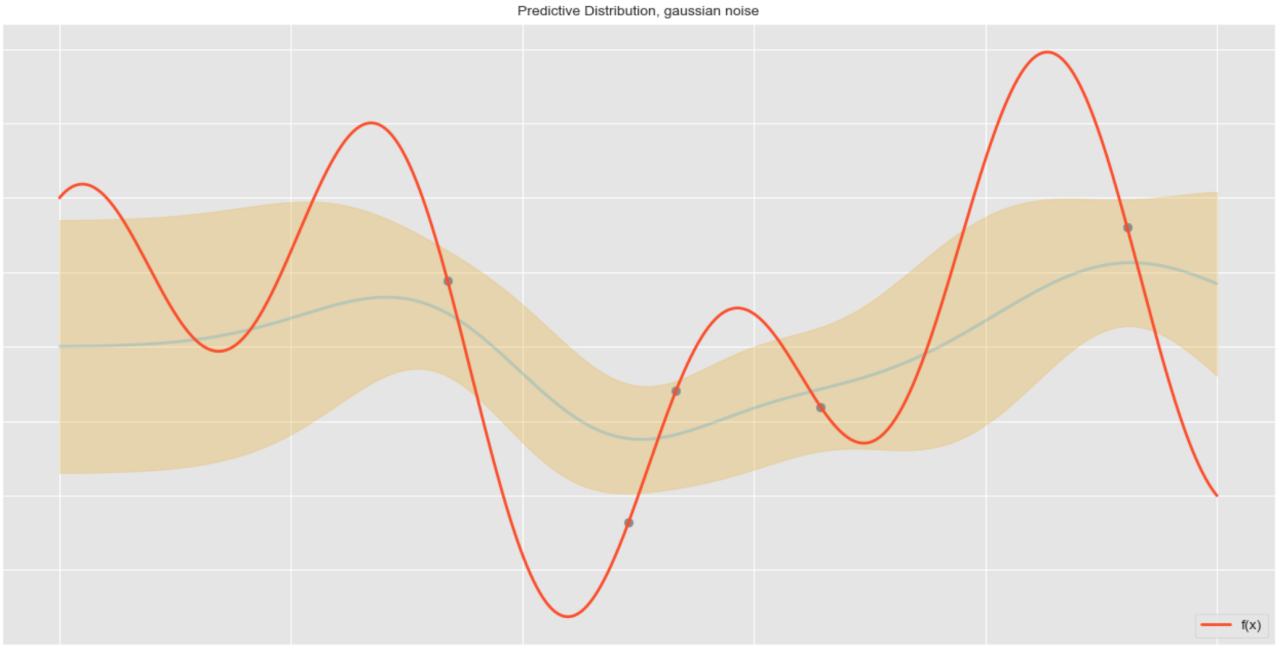


Again, if we assume zero mean in the prior distribution it results

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

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

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



Estimating kernel parameters

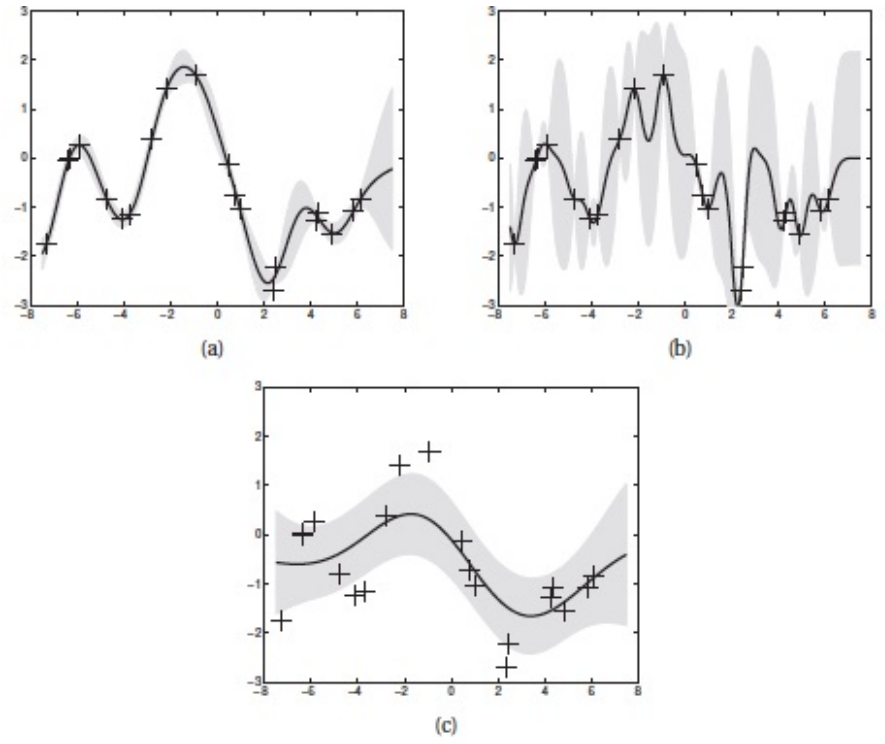
The predictive performance of gaussian processes depends exclusively on the suitability of the chosen kernel.

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

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

$\mathbf{M}$ 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 $(1, 1, 0.1)$, $(0.3, 1.08, 0.00005)$, $(3.0, 1.16, 0.89)$)