Winter Conference in Statistics
Bayesian Machine Learning

Gaussian Process Regression

Mattias Villani

Department of Statistics
Stockholm University
and
Department of Computer and Information Science
Linköping University

- **Bayesian nonlinear regression**
- **Gaussian process regression**

- **Linear regression**

$$y = f(\mathbf{x}) + \epsilon$$

$$f(\mathbf{x}) = \mathbf{x}^T \beta$$

and $\epsilon \sim N(0, \sigma_n^2)$ and iid over observations.

- **Polynomial regression**: $\phi(\mathbf{x}) = (1, x, x^2, x^3, ..., x^k)$:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \beta.$$

- More generally: **splines** with **basis functions**.
- Polynomial and spline models are linear in β. Least squares!

- **Model**: Linear regression for all n observations

$$\underset{n\times 1}{\mathbf{y}} = \underset{n\times q}{\mathbf{X}}\underset{q\times 1}{\beta} + \underset{n\times 1}{\varepsilon} \quad \varepsilon \sim N(0, \sigma_n^2 I_n) \text{ with } \sigma_n \text{ known}$$

- **Prior**

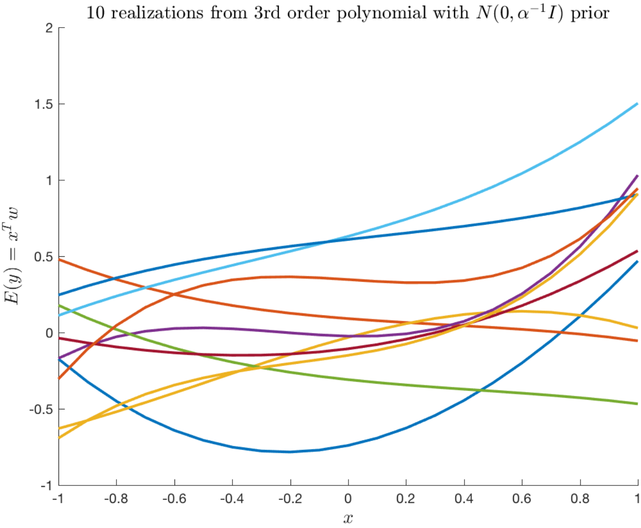
$$\beta \sim N\left(0, \Sigma_p\right)$$

- Common choice (Ridge regression): $\Sigma_p = \lambda^{-1}\mathbf{I}$.
- **Posterior**

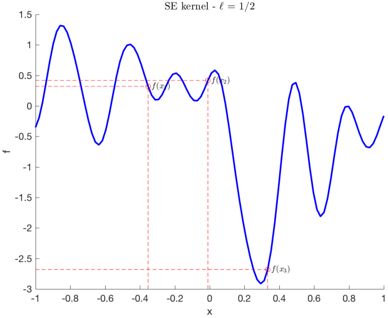
$$\beta|\mathbf{X},\mathbf{y} \sim N\left(\bar{\beta}, \mathbf{A}^{-1}\right)$$
$$\mathbf{A} = \sigma_n^{-2}\mathbf{X}^T\mathbf{X} + \Sigma_p^{-1}$$
$$\bar{\beta} = \sigma_n^{-2}\left(\sigma_n^{-2}\mathbf{X}^T\mathbf{X} + \Sigma_p^{-1}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

- **Posterior precision = Data Precision + Prior Precision**.

A prior on β is really a prior over functions



- **Non-parametric regression**: avoid a parametric form for $f(\cdot)$.
- Treat $f(\mathbf{x})$ as **an unknown parameter for every x**.



- A *new* parameter for *every* $\mathbf{x}$, you must be joking?
- Instead of restricting to linear, impose **smoothness**.

- **Weight space view**
- Restrict attention to a grid of x-values: $x_1, .., x_k$.
- Put a joint prior on the **vector of k function values**

$$f(x_1), ..., f(x_k)$$

- **Function space view**
- Treat f **as an unknown function**.
- Put a prior over a set of functions.

- A GP implies:

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix} \sim N(\mathbf{m}, \mathbf{K})$$

- But how do we specify the $k \times k$ **covariance matrix** $\mathbf{K}$?

$$Cov\left(f(x_p), f(x_q)\right)$$

- **Squared exponential covariance function**

$$Cov\left(f(x_p), f(x_q)\right) = k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{x_p - x_q}{\ell}\right)^2\right)$$

- Nearby x's have highly correlated function ordinates $f(x)$.
- We can compute $Cov\left(f(x_p), f(x_q)\right)$ for *any* x_p and x_q.

Definition

A **Gaussian process** (**GP**) is a collection of random variables, any finite number of which have a multivariate Gaussian distribution.

- A GP is a **probability distribution over functions**.
- A GP is specified by a **mean** and a **covariance function**

$$m(x) = \mathrm{E}\left[f(x)\right]$$

$$k(x, x') = E\left[(f(x) - m(x))\left(f(x') - m(x')\right)\right]$$

  for any two inputs x and x'.

- A **Gaussian process** is denoted by

$$f(x) \sim \mathrm{GP}\left(m(x), k(x, x')\right)$$

- $f(x) \sim \mathrm{GP}$ encodes **prior beliefs** about the unknown $f(\cdot)$.

- Let $r = \|x - x'\|$.
- **Squared exponential** (**SE**) kernel ($\ell > 0$, $\sigma_f > 0$)

$$K_{SE}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2\ell^2}\right)$$

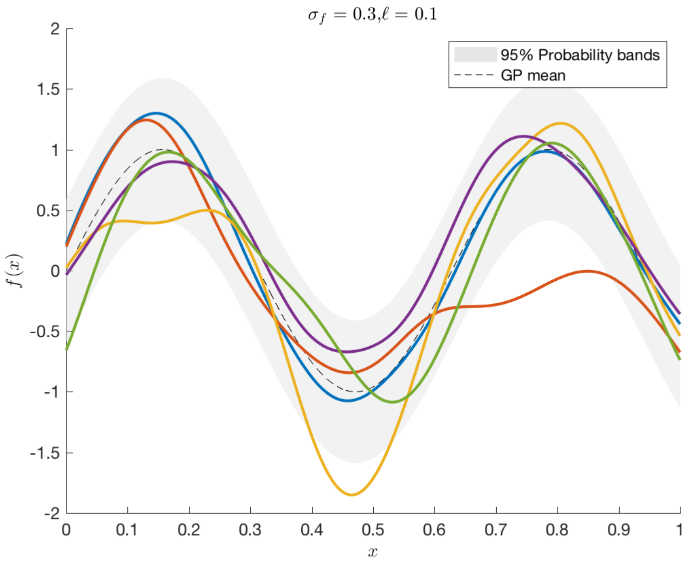
- **Matérn** kernel ($\ell > 0$, $\sigma_f > 0$, $\nu > 0$)

$$K_{Matern}(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}r}{\ell}\right)$$

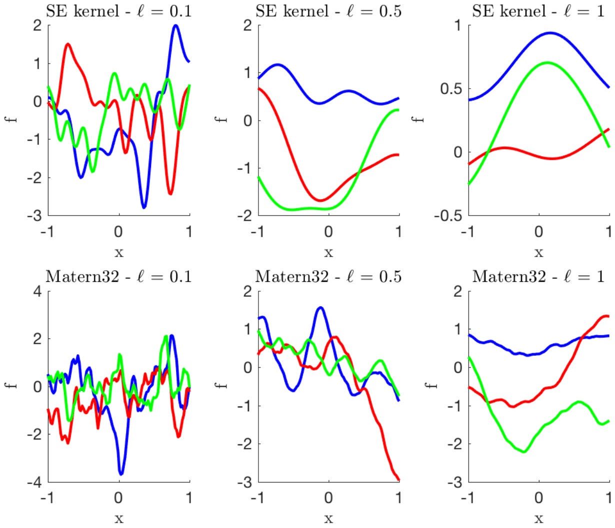
- **Simulate draw** from $f(x) \sim \mathrm{GP}(m(x), k(x, x'))$ by:
  - form a grid $\mathbf{x}_* = (x_1, \ldots, x_n)$
  - simulate function values from multivariate normal:

$$f(\mathbf{x}_*) \sim N(m(\mathbf{x}_*), K(\mathbf{x}_*, \mathbf{x}_*))$$

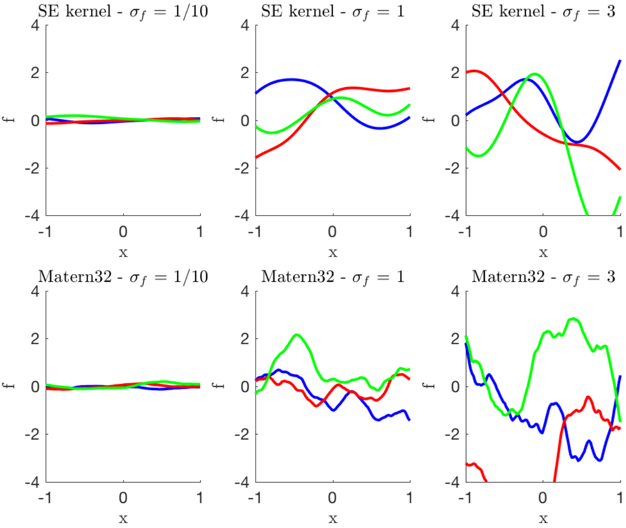
Simulating a GP



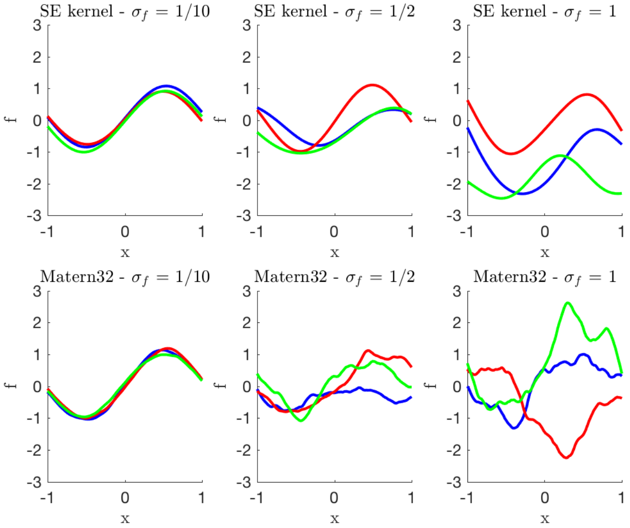
The length scale ℓ determines the smoothness



The scale factor σ_f determines the variance



The mean can be sin(3x). Or whatever.



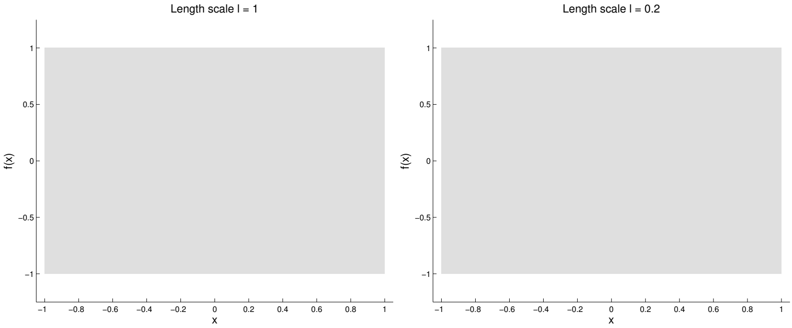
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix} \sim N(\mathbf{m}, \mathbf{K})$$

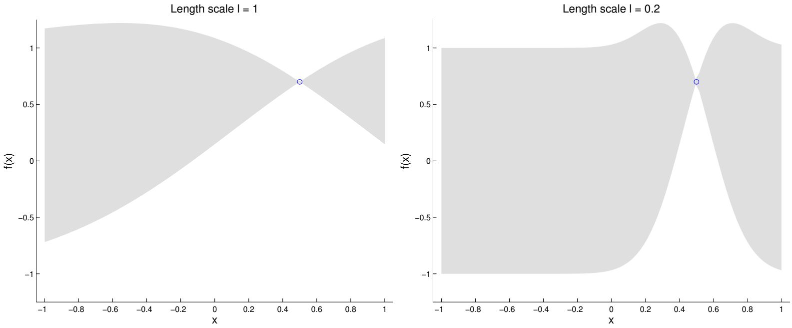
- More intuition from the conditional decomposition

$$\begin{aligned} p\left(f(x_1), f(x_2), ..., f(x_k)\right) &= p\left(f(x_1)\right) p\left(f(x_2)|f(x_1)\right) \cdots \\ &\times p\left(f(x_k)|f(x_1), ..., f(x_{k-1})\right) \end{aligned}$$

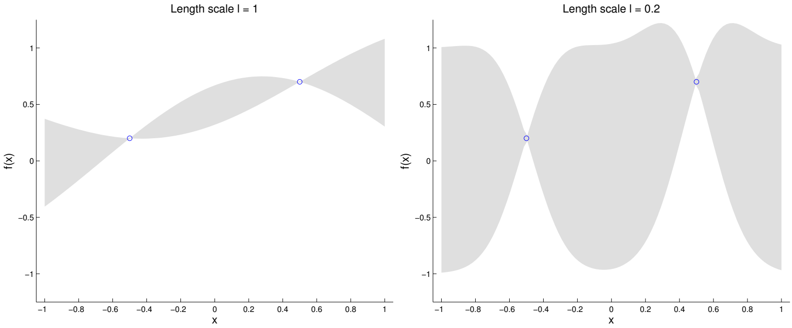
Simulating from $p(f(x_1))$



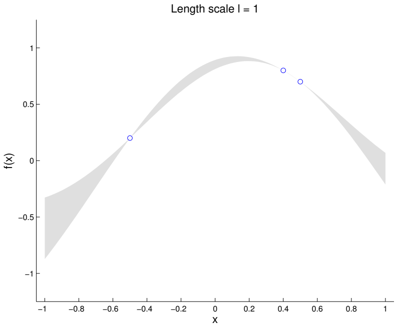
SIMULATING FROM $p\left(f(x_2)|f(x_1)\right)$

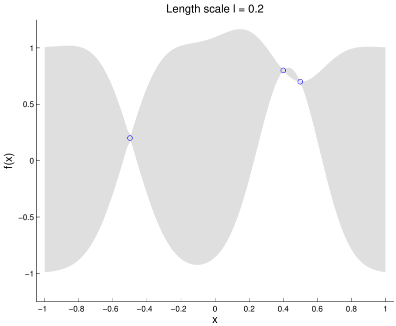


SIMULATING FROM $p\left(f(x_3)|f(x_1), f(x_2)\right)$



Simulating from $p(f(x_4)|f(x_1), f(x_2), f(x_3))$





- **Model**

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon \overset{iid}{\sim} N(0, \sigma_n^2)$$

- **Prior**

$$f(x) \sim GP\left(0, k(x, x')\right)$$

- **Observed**: $\mathbf{x} = (x_1, ..., x_n)^T$ and $\mathbf{y} = (y_1, ..., y_n)^T$.
- **Goal**: posterior of $f(\cdot)$ over a grid of x-values: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.
- **Posterior**

$$\begin{aligned}
\mathbf{f}_*|\mathbf{x}, \mathbf{y}, \mathbf{x}_* &\sim N\left(\bar{\mathbf{f}}_*, \operatorname{cov}(\mathbf{f}_*)\right) \\
\bar{\mathbf{f}}_* &= K(\mathbf{x}_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I\right]^{-1}\mathbf{y} \\
\operatorname{cov}(\mathbf{f}_*) &= K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I\right]^{-1} K(\mathbf{x}, \mathbf{x}_*)
\end{aligned}$$

- Idea: obtain joint $p(\mathbf{y}, \mathbf{f}_*)$ and then $p(\mathbf{f}_*|\mathbf{y})$ by conditioning.
- **Model**

$$y_i = f(\mathrm{x}_i) + \varepsilon_i, \quad \varepsilon \overset{iid}{\sim} N(0, \sigma_n^2)$$

- **Prior**

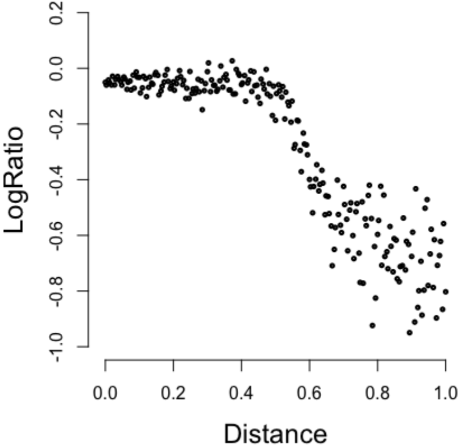
$$f(x) \sim GP\left(0, k(x, x')\right)$$

- Joint distribution of $(\mathbf{y}, \mathbf{f}_*)$

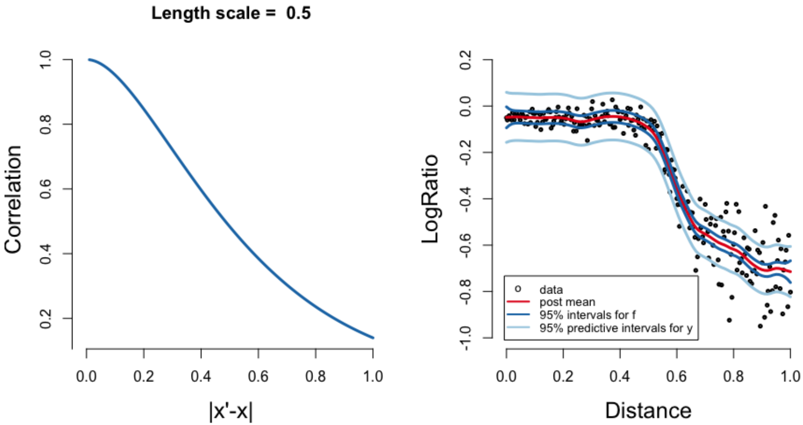
$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathrm{N}\left[\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I & K(\mathbf{x}, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathbf{x}) & K(\mathbf{x}_*, \mathbf{x}_*) \end{pmatrix}\right]$$

- Result: conditional distributions from multivariate normal are normal.

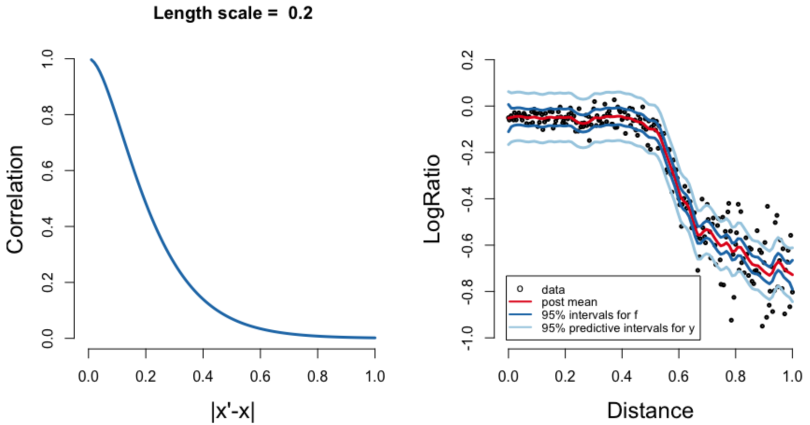
Example - LIDAR data



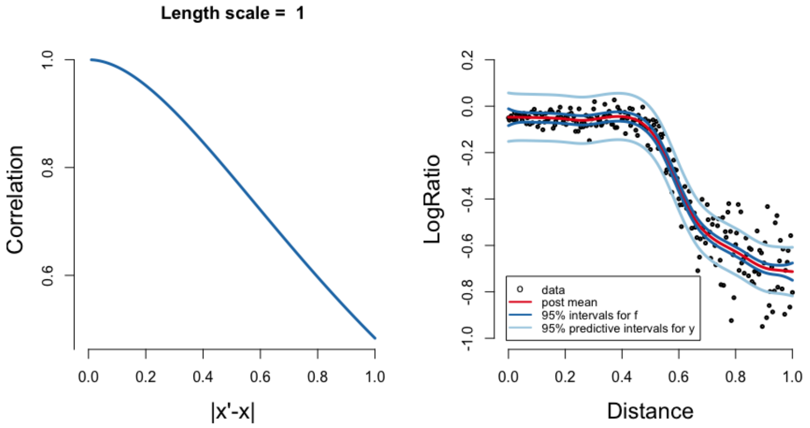
GP FIT TO LIDAR DATA $\ell = 0.5, \sigma_f = 0.5, \sigma_n = 0.05$

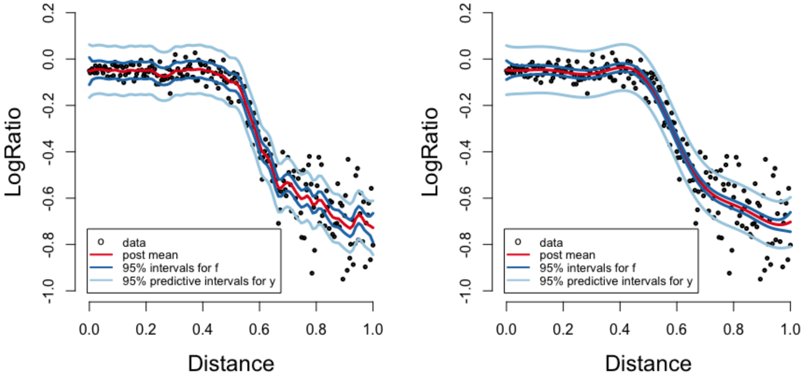


GP fit to LIDAR data $\ell = 0.2, \sigma_f = 0.5, \sigma_n = 0.05$



GP fit to LIDAR data $\ell = 1, \sigma_f = 0.5, \sigma_n = 0.05$




LogRatio
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
0.0
0.2
0.4
0.6
0.8
1.0
Distance
data
post mean
95% intervals for f
95% predictive intervals for y
LogRatio
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
0.0
0.2
0.4
0.6
0.8
1.0
Distance
data
post mean
95% intervals for f
95% predictive intervals for y

- LIDAR data is clearly heteroscedastic.

- **Heteroscedastic GP regression**

$$y = f(x) + \exp\left[g(x)\right]\epsilon, \quad \epsilon \sim N(0, I_n)$$

with **mean function**

$$f \sim GP\left[0, k_f(x, x')\right]$$

a priori independent of log **variance function**

$$g \sim GP\left[0, k_g(x, x')\right]$$

- Posterior is not tractable anymore.
- Idea: sample from $p(\mathbf{f}, \mathbf{g}|\mathbf{y}, \mathbf{X}) = p(\mathbf{f}|\mathbf{g}, \mathbf{y}, \mathbf{X})p(\mathbf{g}|\mathbf{y}, \mathbf{X})$.
- $p(\mathbf{f}|\mathbf{g}, \mathbf{y}, \mathbf{X})$ is normal and $p(\mathbf{g}|\mathbf{y}, \mathbf{X})$ in closed form.
- MCMC or slice sampling for $p(\mathbf{g}|\mathbf{y}, \mathbf{X})$.

- Kernel depends on **hyperparameters** $\theta = (\sigma_f, \ell)^T$. Example

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\ell^2}\right)$$

- Common: maximize the **marginal likelihood** wrt θ:

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{X}, \mathbf{f}, \theta) p(\mathbf{f}|\mathbf{X}, \theta) d\mathbf{f}$$

  $\mathbf{f} = f(\mathbf{X})$ is a vector of function values in the training data.
- For **Gaussian process regression**:

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y} - \frac{1}{2}\log\left|K + \sigma_n^2 I\right| - \frac{n}{2}\log(2\pi)$$

- Proper **Bayesian inference for hyperparameters**

$$p(\theta|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \theta) p(\theta).$$

- Choice of kernel family by Bayesian model inference. For kernel $K_i \in \mathcal{K}$: $p(K_i|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \theta, K_i) p(K_i)$.