Machine Learning

Lecture 8 - Gaussian process regression and classification

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University

mattiasvillani.com @matvil mattiasvillani

Lecture overview

- Bayesian inference
- Gaussian process regression
- Gaussian process classification

Bayesian inference

- Parametric model $p(x|\boldsymbol{\theta})$.
- **Likelihood**: $p(x_1, \ldots, x_n|\boldsymbol{\theta})$.
- Bayesian inference uses Bayes' theorem to combine
  - data information (likelihood)
  - other information (prior)
- **Prior** distribution $p(\boldsymbol{\theta})$.
- **Subjective probability** for **unknown** quantities.
- **Posterior distribution**

$$p(\boldsymbol{\theta}|x_1, \ldots, x_n) \propto p(x_1, \ldots, x_n|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

Posterior $\propto$ **Likelihood** $\times$ **Prior**

Normal data, known variance - normal prior

- **Model**

$$x_1, ..., x_n | \theta, \sigma^2 \overset{\text{iid}}{\sim} N(\theta, \sigma^2).$$

- **Prior**

$$\theta \sim N(\mu_0, \tau_0^2)$$

- **Posterior**

$$\begin{aligned} p(\theta | x_1, ..., x_n) &\propto p(x_1, ..., x_n | \theta, \sigma^2) p(\theta) \\ &\propto N(\theta | \mu_n, \tau_n^2), \end{aligned}$$

where

$$\frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2},$$

$$\mu_n = w\bar{x} + (1 - w)\mu_0,$$

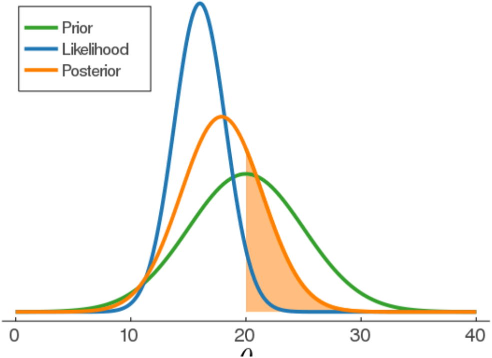
and

$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}.$$

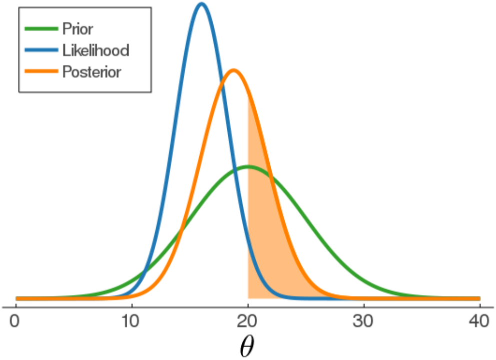
Download speed

- **Problem**: My internet provider promises an average download speed of at least 20 Mbit/sec. Are they lying?
- **Data**: $x = (22.42, 34.01, 35.04, 38.74, 25.15)$ Mbit/sec.
- **Model**: $X_1, ..., X_5 \sim N(\theta, \sigma^2)$.
- Assume $\sigma = 5$ (measurements can vary $\pm$10MBit with 95% probability)
- My **prior**: $\theta \sim N(20, 5^2)$.

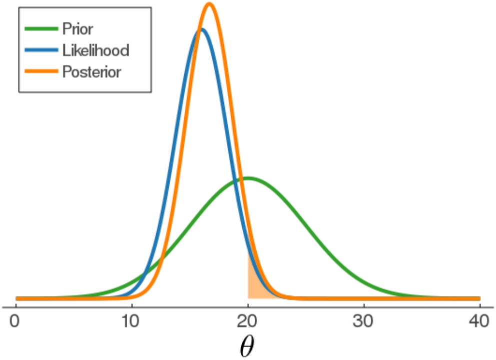
Download speed n=1



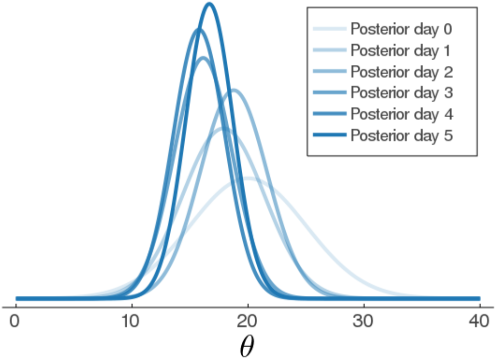
Download speed n=2



Download speed n=5



Bayesian updating



Linear regression with known variance

- The linear regression model in **matrix form**

$$\underset{(n\times 1)}{\mathrm{y}} = \underset{(n\times k)(k\times 1)}{\mathrm{X}\beta} + \underset{(n\times 1)}{\varepsilon}, \quad \varepsilon_i \overset{\text{iid}}{\sim} N(0,\sigma^2)$$

- **Prior** for β

$$\beta \sim N\left(0, \sigma^2\Omega_0^{-1}\right)$$

- **Posterior**

$$\beta|\mathrm{y},\mathrm{X} \sim N\left[\mu_n, \sigma^2\Omega_n^{-1}\right]$$

$$\mu_n = \left(\mathrm{X}^\top\mathrm{X} + \Omega_0\right)^{-1}\mathrm{X}^\top\mathrm{X}\hat{\beta}$$

$$\Omega_n = \mathrm{X}^\top\mathrm{X} + \Omega_0$$

- **Posterior mean** estimate μ_n is a shrunken version of least squares estimate $\hat{\beta} = (\mathrm{X}^\top\mathrm{X})\mathrm{X}^\top\mathrm{y}$.
- Prior acts as **regularization**. $\Omega_0 = \lambda I$ gives **Ridge**.

Nonlinear regression

- **Linear regression**

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

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

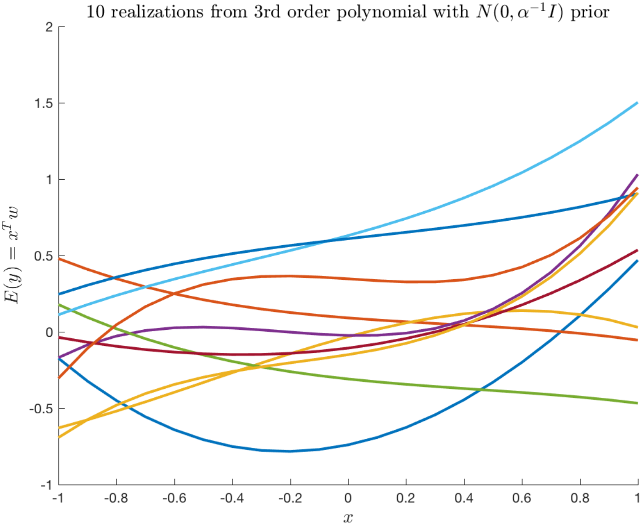
and $\epsilon \sim N(0, \sigma_n^2)$ and iid over observations.

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

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

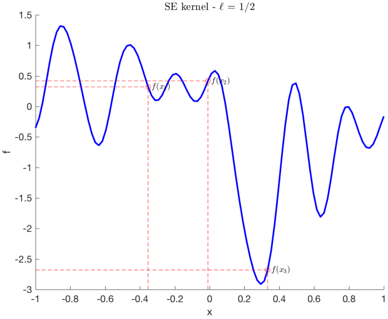
- More generally: **splines** with **basis functions**.

A prior on β is really a prior over functions



Non-parametric regression

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



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

Two views on GPs

- **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.

Gaussian process and its kernel

- A GP implies:

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

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

$$Cov(f(x_p), f(x_q))$$

- **Squared exponential covariance function**

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

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

Gaussian processes

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**

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

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

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 $f(\cdot)$.

Gaussian processes

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

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

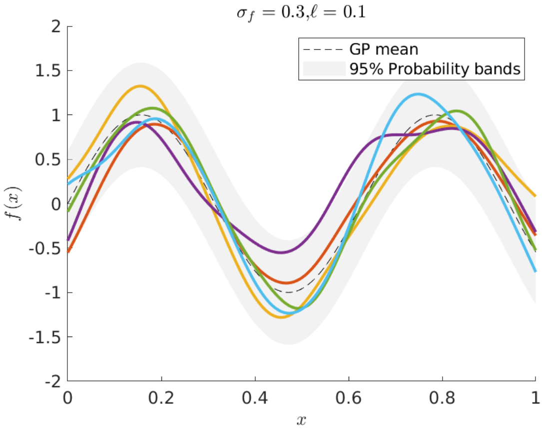
- **Matérn** kernel ($\ell > 0$, $\sigma_f > 0$, $\nu > 0$)

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

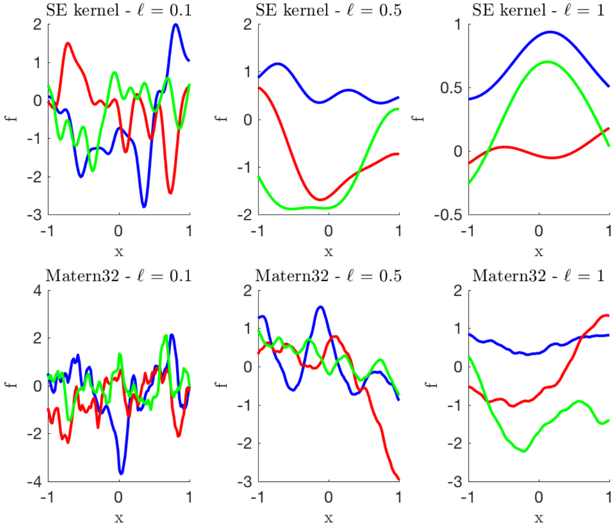
- **Simulate draw** from $f(x) \sim \mathrm{GP}(m(x), k(x, x'))$ by:
  - form a grid $\mathrm{x}_* = (x_1, ..., x_n)$
  - simulate function values from multivariate normal:

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

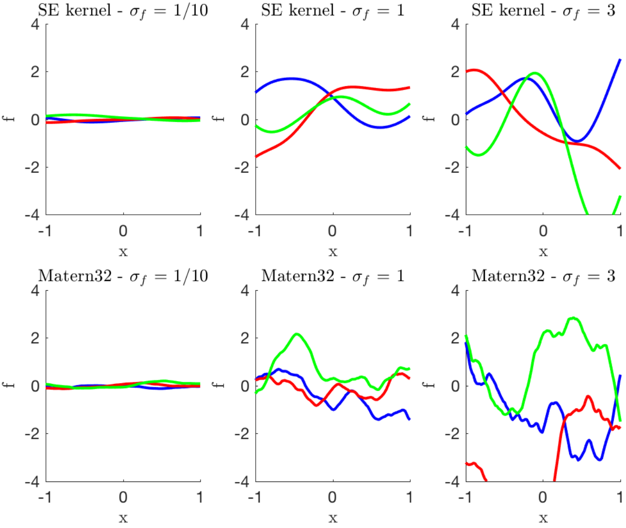
Simulating a GP



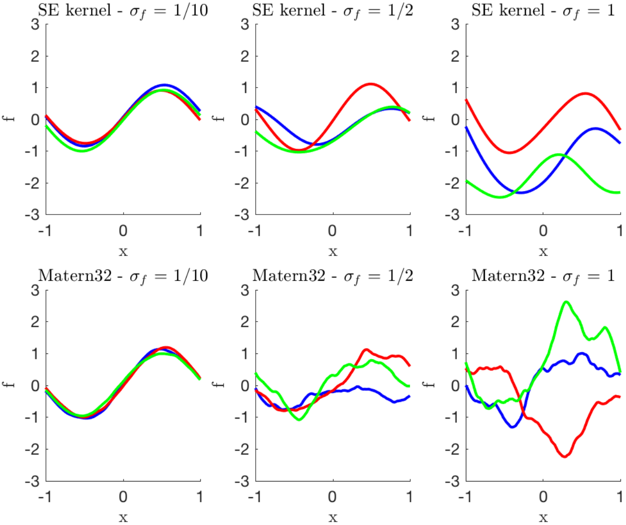
The length scale ℓ determines the smoothness



The scale factor σ_f determines the variance



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



Sequential simulation of GPs

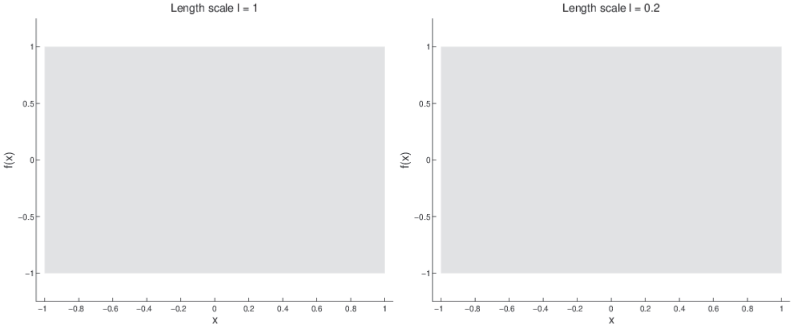
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

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

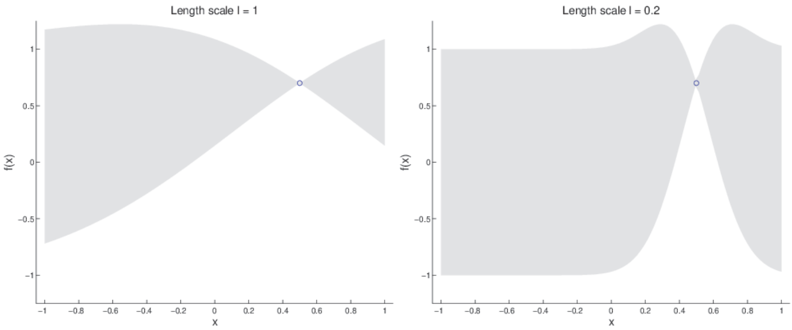
- More intuition from the conditional decomposition

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

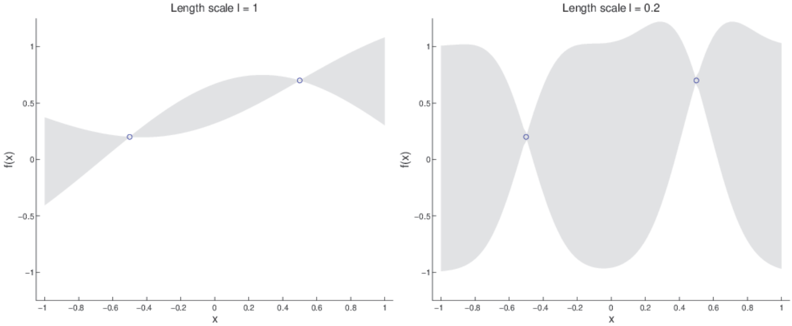
Simulating from $p(f(x_1))$



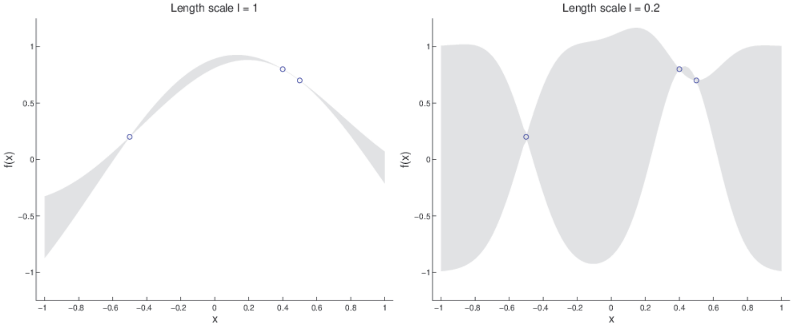
Simulating from $p(f(x_2)|f(x_1))$



Simulating from $p\left(f\left(x_3\right)\middle|f\left(x_1\right), f\left(x_2\right)\right)$



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



The posterior for a Gaussian Process Regression

- **Model**

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

- **Prior**

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

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

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

Scetch for proof of posterior

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

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

- **Prior**

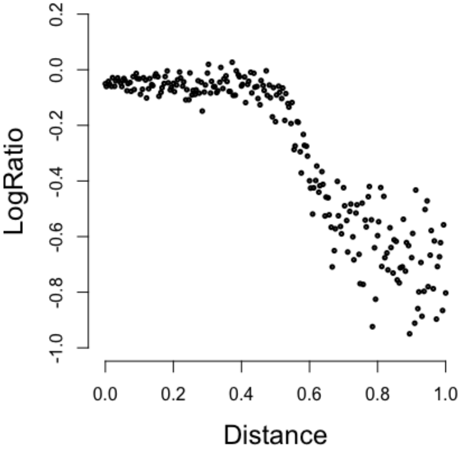
$$f(x) \sim GP\left(0, k(x, x')\right)$$

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

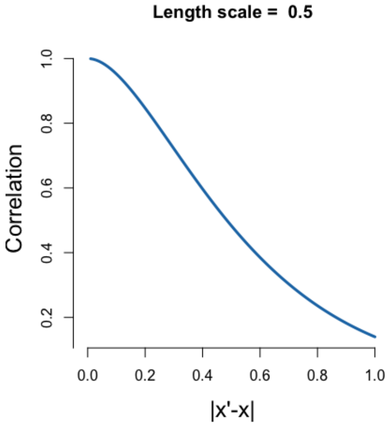
$$\begin{pmatrix} \mathrm{y} \\ \mathrm{f}_* \end{pmatrix} \sim \mathrm{N}\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(\mathrm{x}, \mathrm{x}) + \sigma_n^2 I & K(\mathrm{x}, \mathrm{x}_*) \\ K(\mathrm{x}_*, \mathrm{x}) & K(\mathrm{x}_*, \mathrm{x}_*) \end{pmatrix}\right]$$

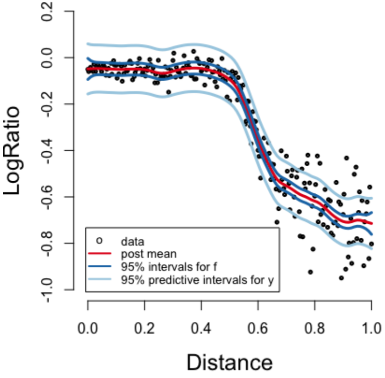
- Result: conditional distributions from multivariate normal are normal.

Example - LIDAR data

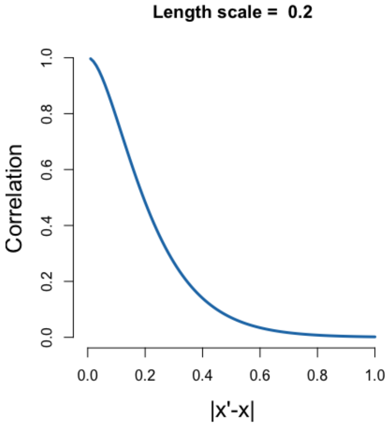


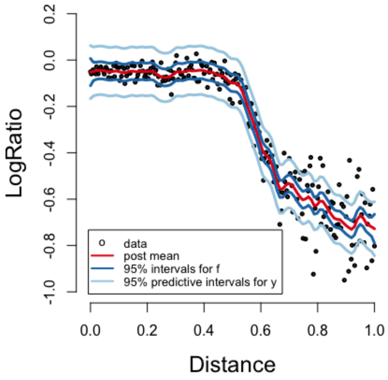
GP fit to LIDAR data $\ell = 0.5, \sigma_f = 0.5, \sigma_n = 0.05$



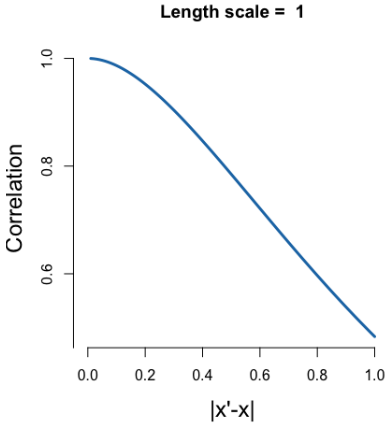


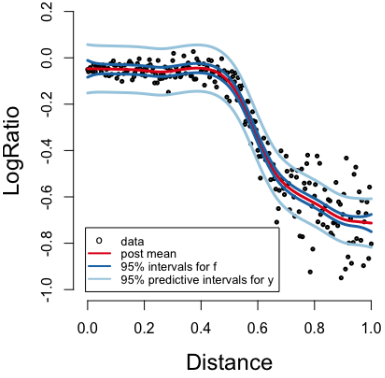
GP fit to LIDAR data $\ell = 0.2, \sigma_f = 0.5, \sigma_n = 0.05$



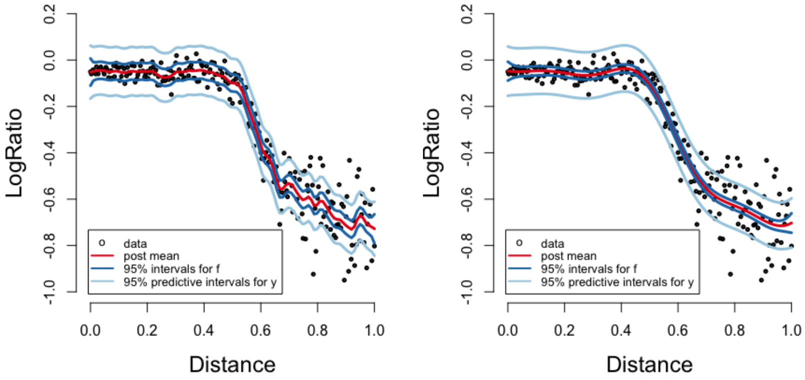


GP fit to LIDAR data $\ell = 1, \sigma_f = 0.5, \sigma_n = 0.05$





Matern32 vs SquaredExp for $\ell = 0.2$



Inference for the hyperparameters

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

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

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

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

$\mathrm{f} = f(\mathrm{X})$ is a vector of function values in the training data.

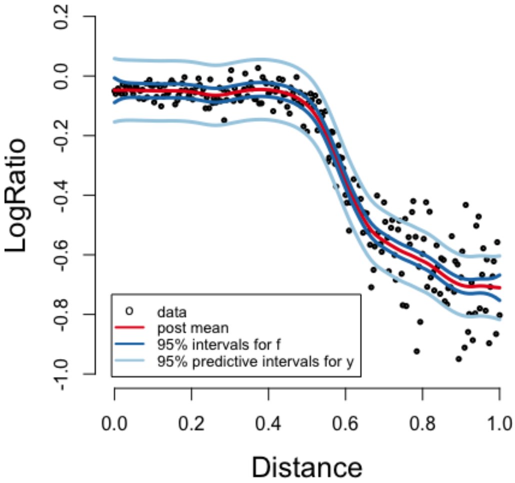
- For **Gaussian process regression**:

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

- Proper **Bayesian inference** **for hyperparameters**

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

GP fit LIDAR $\ell_{opt} = 0.61, \sigma_{f,opt} = 0.44, \sigma_n = 0.05$



GP Classification

- **Binary** or multi-class **response**. Aim: $\Pr(y_i = 1|\mathrm{x}_i)$.
- **Logistic regression**

$$\Pr(y_i = 1|\mathrm{x}_i) = \lambda(\mathrm{x}_i^T\beta), \text{ where } \lambda(z) = \frac{1}{1+\exp(-z)}.$$

- $\lambda(z)$ 'squashes' the linear prediction x^T¿ $\in \mathbb{R}$ into $[0, 1]$.
- **Linear decision boundaries** because of linear predictor $\mathrm{x}^T\beta$.
- **GP classification**: replace $\mathrm{x}^T\beta$ by $f(\mathrm{x})$ where

$$f \sim \mathrm{GP}(0, k(\mathrm{x}, \mathrm{x}'))$$

  and squash f through logistic function

$$\Pr(y = 1|\mathrm{x}) = \lambda(f(\mathrm{x}))$$

- Nonparametric **flexible decision boundaries**.

GP Classification on simulated data