

# 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.

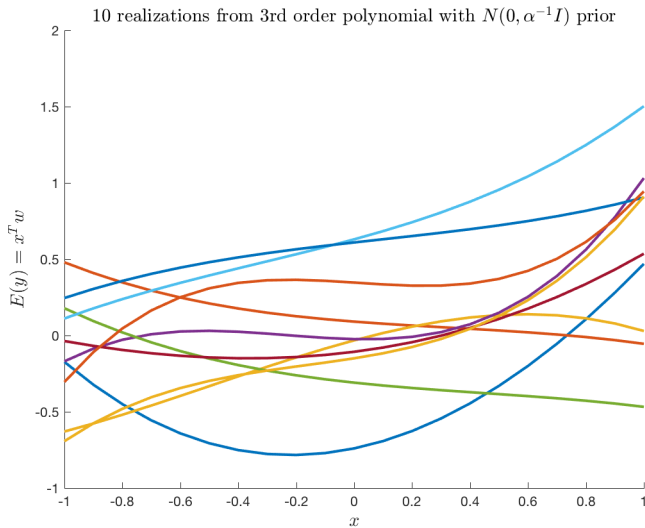
## ■ Polynomial regression: $\phi(\mathbf{x}) = (1, x, x^2, x^3, \dots, x^k)$ :

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

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

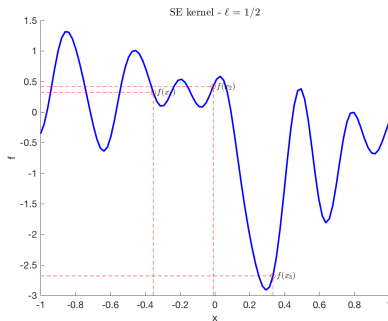
■ Polynomial and spline models are linear in  $\beta$ . Least squares!

# A PRIOR ON $\beta$ IS REALLY A PRIOR OVER FUNCTIONS



# NON-PARAMETRIC REGRESSION

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



- 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:  $x_1, \dots, x_k$ .
- Put a joint prior on the **vector of  $k$  function values**

$$f(x_1), \dots, f(x_k)$$

---

## ■ 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**  $\mathbf{K}$ ?

$$\text{Cov}(f(x_p), f(x_q))$$

- **Squared exponential covariance function**

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

- Nearby  $x$ 's have highly correlated function ordinates  $f(x)$ .
- We can compute  $\text{Cov}(f(x_p), f(x_q))$  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**

$$m(x) = E[f(x)]$$

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

for any two inputs  $x$  and  $x'$ .

- A **Gaussian process** is denoted by

$$f(x) \sim \text{GP}(m(x), k(x, x'))$$

- $f(x) \sim \text{GP}$  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 ( $\ell > 0, \sigma_f > 0, \nu > 0$ )

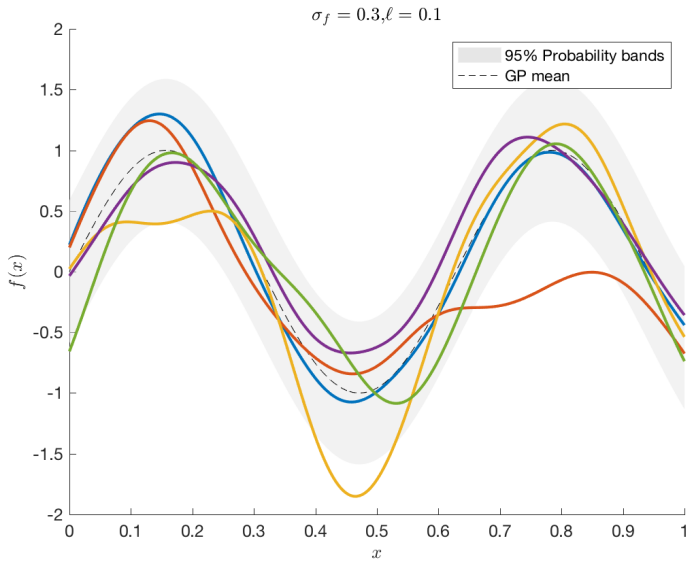
$$K_{Matern}(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}r}{\ell}\right)$$

■ **Simulate draw** from  $f(x) \sim \text{GP}(m(x), k(x, x'))$  by:

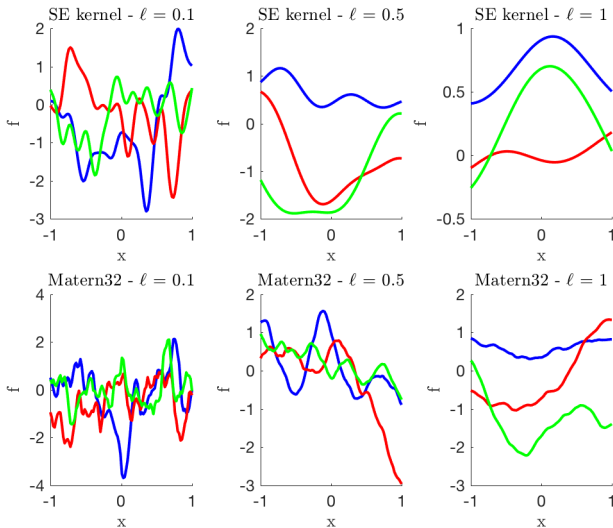
- form a grid  $\mathbf{x}_* = (x_1, \dots, x_n)$
- simulate function values from multivariate normal:

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

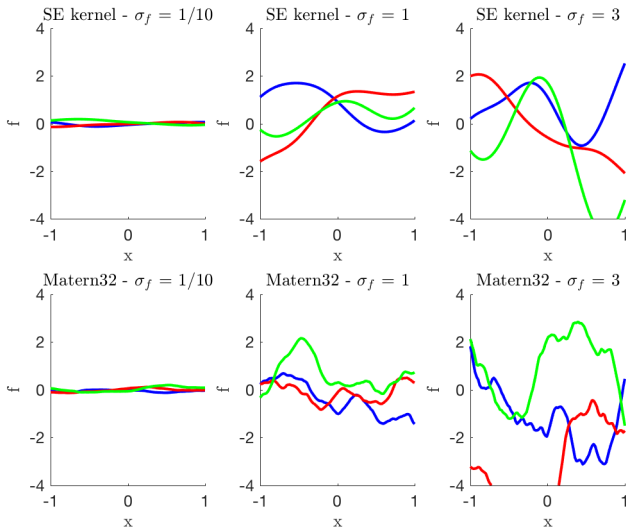
# SIMULATING A GP



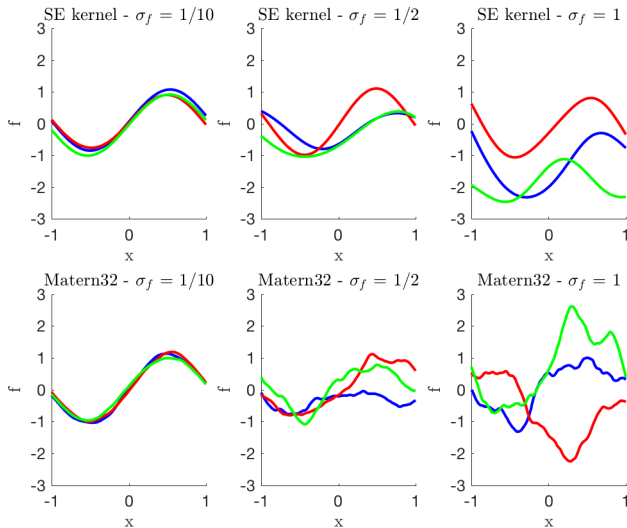
# THE LENGTH SCALE $\ell$ DETERMINES THE SMOOTHNESS



# THE SCALE FACTOR $\sigma_f$ DETERMINES THE VARIANCE



# THE MEAN CAN BE $\sin(3x)$ . OR WHATEVER.



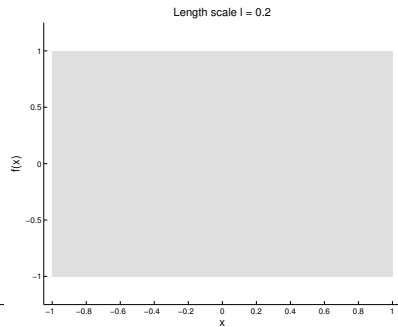
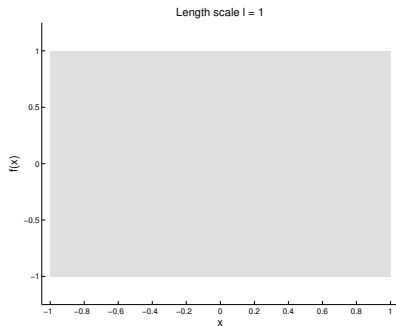
- The joint way: Choose a grid  $x_1, \dots, x_k$ . Simulate the  $k$ -vector

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

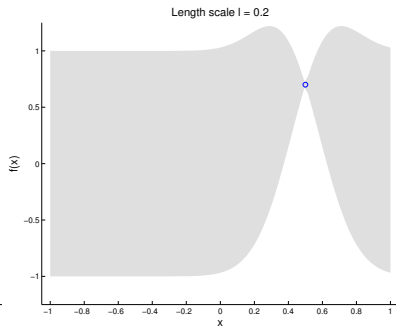
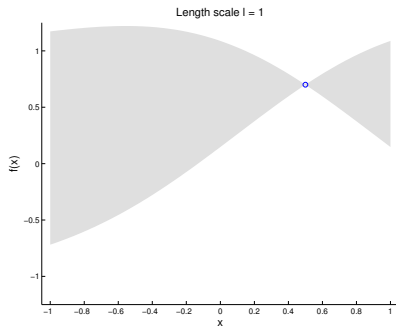
- More intuition from the conditional decomposition

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

# SIMULATING FROM $p(f(x_1))$

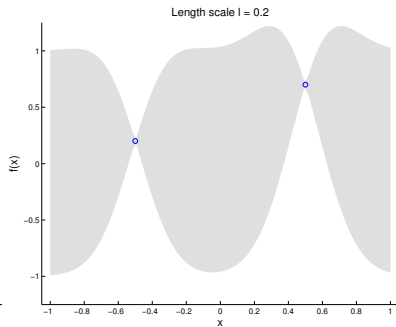
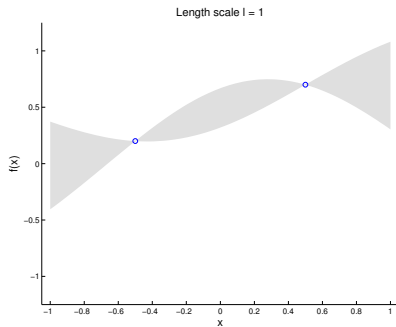


# SIMULATING FROM $p(f(x_2)|f(x_1))$

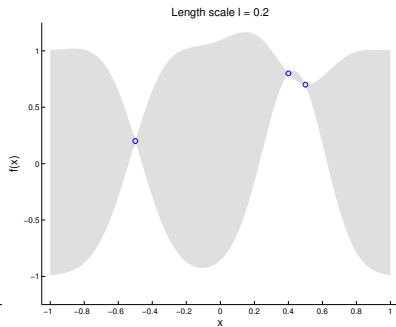
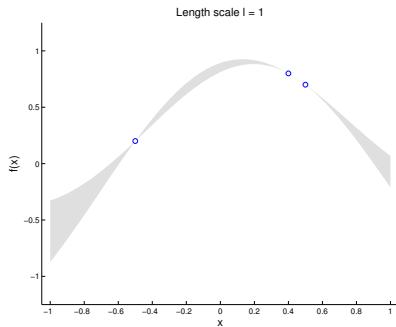




# SIMULATING FROM $p(f(x_3)|f(x_1), f(x_2))$



# SIMULATING FROM $p(f(x_4)|f(x_1), f(x_2), f(x_3))$



# THE POSTERIOR FOR A GAUSSIAN PROCESS REGRESSION

## ■ Model

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

## ■ Prior

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

■ **Observed:**  $\mathbf{x} = (x_1, \dots, x_n)^T$  and  $\mathbf{y} = (y_1, \dots, y_n)^T$ .

■ **Goal:** posterior of  $f(\cdot)$  over a grid of  $x$ -values:  $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$ .

## ■ Posterior

$$\mathbf{f}_* | \mathbf{x}, \mathbf{y}, \mathbf{x}_* \sim N(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

$$\bar{\mathbf{f}}_* = K(\mathbf{x}_*, \mathbf{x}) [K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} \mathbf{y}$$

$$\text{cov}(\mathbf{f}_*) = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x}) [K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} K(\mathbf{x}, \mathbf{x}_*)$$

# SCETCH FOR PROOF OF POSTERIOR

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

- **Model**

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

- **Prior**

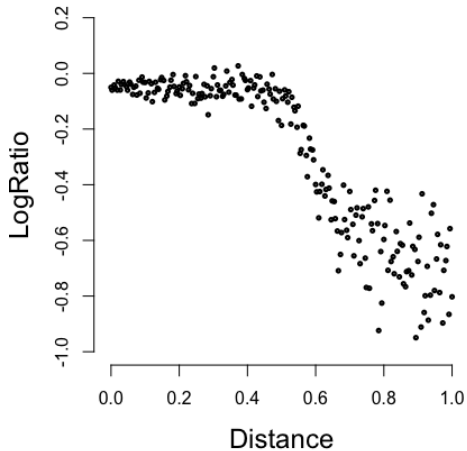
$$f(\mathbf{x}) \sim GP(0, k(\mathbf{x}, \mathbf{x}'))$$

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

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

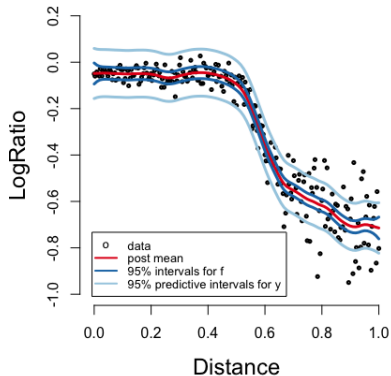
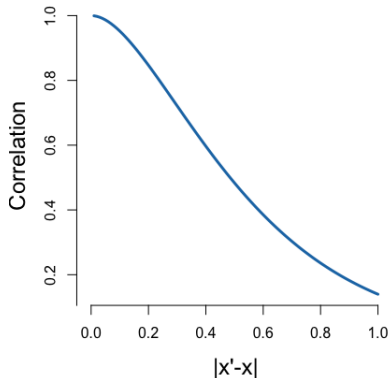
- Result: conditional distributions from multivariate normal are normal.

## EXAMPLE - LIDAR DATA



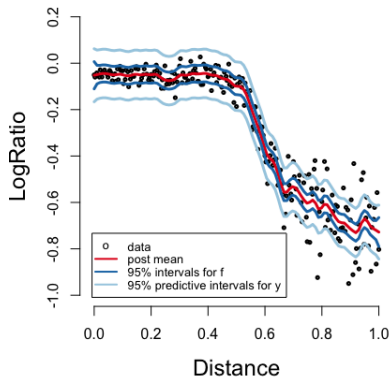
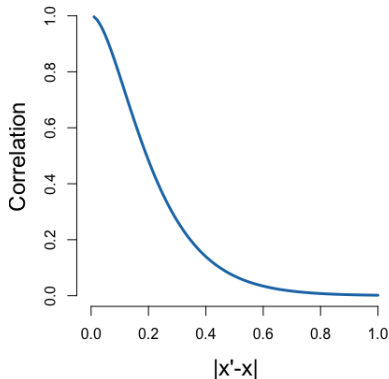
# GP FIT TO LIDAR DATA $\ell = 0.5, \sigma_f = 0.5, \sigma_n = 0.05$

Length scale = 0.5

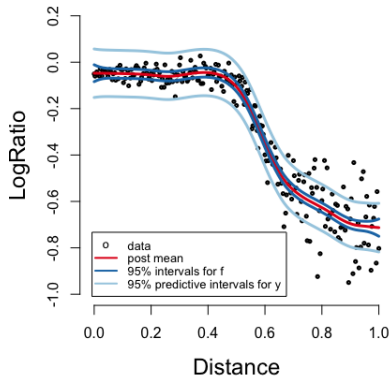
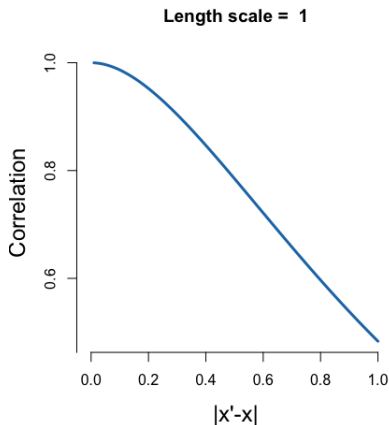


# GP FIT TO LIDAR DATA $\ell = 0.2, \sigma_f = 0.5, \sigma_n = 0.05$

Length scale = 0.2

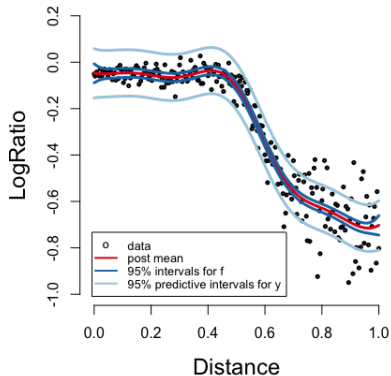
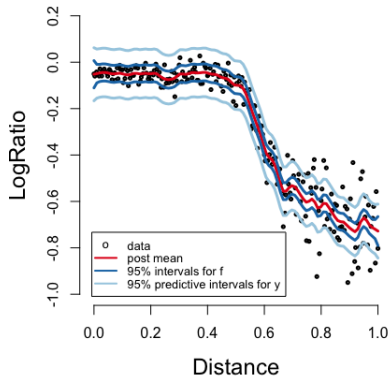


# GP FIT TO LIDAR DATA $\ell = 1, \sigma_f = 0.5, \sigma_n = 0.05$





# MATERN32 VS SQUARED $\bar{\text{EXP}}$ FOR $\ell = 0.2$



# INFERENCE FOR THE HYPERPARAMETERS

- 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  $\theta$ :

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

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

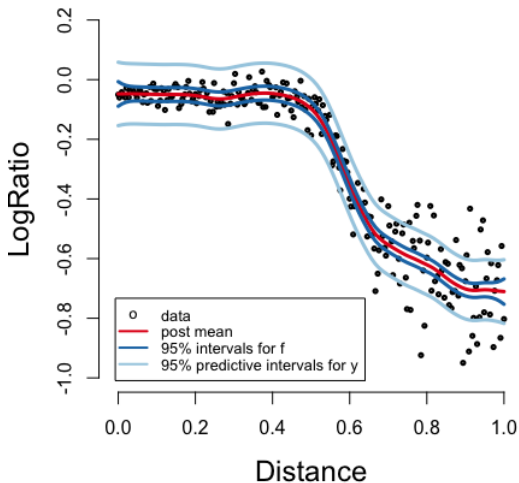
- For **Gaussian process regression**:

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

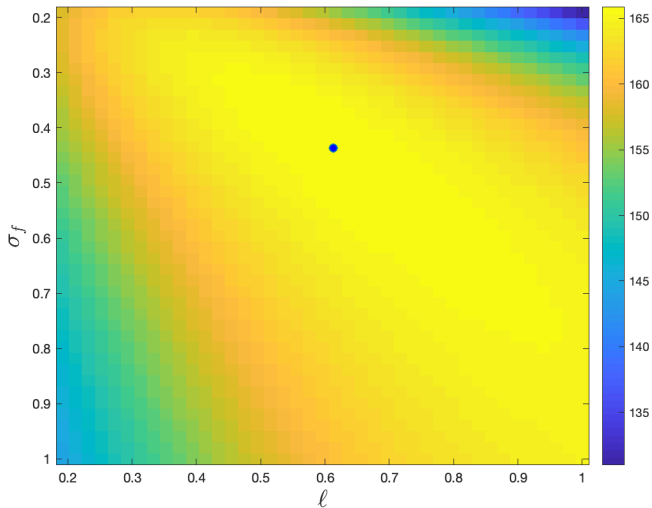
- Proper **Bayesian inference for hyperparameters**

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

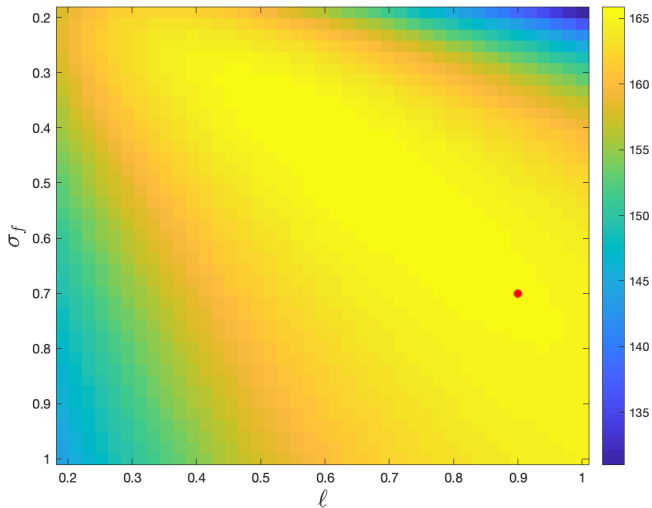
- Choice of kernel family by Bayesian model inference. For kernel  $K_i \in \mathcal{K}$ :  $p(K_i|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \theta, K_i) p(K_i)$ .



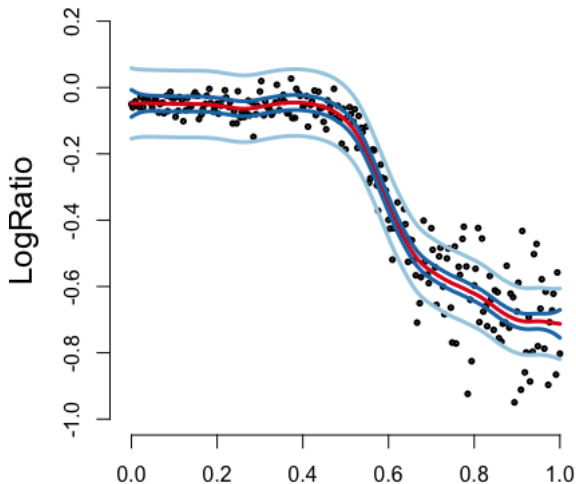
# LOG MARGINAL LIKELIHOOD SURFACE $\sigma_n = 0.05$



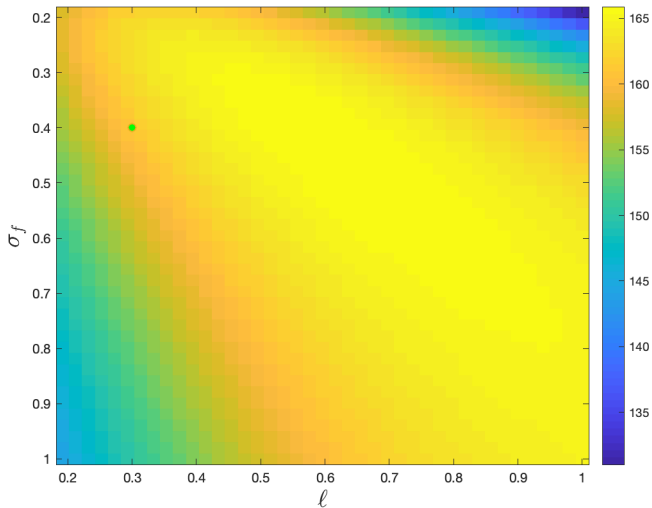
# LOG MARGINAL LIKELIHOOD SURFACE $\sigma_n = 0.05$



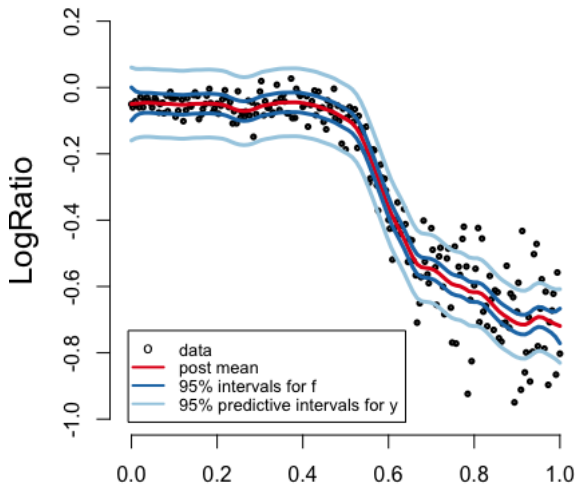
# GP FIT TO LIDAR DATA $\ell = 0.9, \sigma_f = 0.7, \sigma_n = 0.05$



# LOG MARGINAL LIKELIHOOD SURFACE $\sigma_n = 0.05$



# GP FIT TO LIDAR DATA $\ell = 0.3, \sigma_f = 0.4, \sigma_n = 0.05$





# HETEROSCEDASTIC GP REGRESSION

- LIDAR data is clearly heteroscedastic.

- **Heteroscedastic GP regression**

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

with **mean function**

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

a priori independent of log **variance function**

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

- 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  $p(\mathbf{g} | \mathbf{y}, \mathbf{X})$ .