Advanced Bayesian Learning

Gaussian Process Regression and Classification - Lecture 1

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University

Course overview

- **Four topics**
  - **Gaussian Process Regression and Classification**
  - **Bayesian Nonparametrics**
  - **Variational Inference**
  - **Bayesian Regularization**

- **Examination**
  - Individual Lab/Exercise for each topic
  - Deadline for submission: day before new topic starts.
  - Extra deadline for all four topics: Sept 15, 2024.

Topic overview

- Gaussian Process Regression
- Gaussian Process Classification

Nonlinear regression

- **Linear regression**

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

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

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

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

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

- More generally: **splines** with **basis functions**.
- Example: **thin plate splines** with knots $\kappa_1, \ldots, \kappa_N$ in x-space

$$\phi_k(\mathrm{x}) = \ln\left(\|\mathrm{x} - \kappa_k\|\right) \|\mathrm{x} - \kappa_k\|^2$$

Recap: Bayesian linear regression

- **Prior**

$$\mathrm{w} \sim N\left(0, \Sigma_p\right)$$

- **Posterior** [X is $D \times n$]

$$\mathrm{w}|\mathrm{X},\mathrm{y} \sim N\left(\bar{\mathrm{w}}, \mathrm{A}^{-1}\right)$$

$$\mathrm{A} = \sigma_n^{-2}\mathrm{X}\mathrm{X}^T + \Sigma_p^{-1}$$

$$\bar{\mathrm{w}} = \sigma_n^{-2}\mathrm{A}^{-1}\mathrm{X}\mathrm{y} = \left(\mathrm{X}\mathrm{X}^T + \sigma_n^2\Sigma_p^{-1}\right)^{-1}\mathrm{X}\mathrm{y}$$

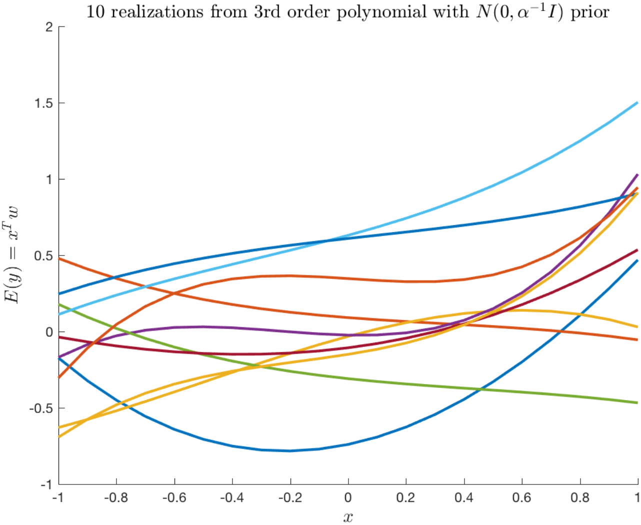
- **Predictive density for mean $f(\mathrm{x}_*) = \mathrm{x}_*^T\mathrm{w}$** at new x_*

$$f(\mathrm{x}_*)|\mathrm{x}_*, \mathrm{X}, \mathrm{y} \sim N\left(\mathrm{x}_*^T\bar{\mathrm{w}}, \mathrm{x}_*^T\mathrm{A}^{-1}\mathrm{x}_*\right)$$

- **Predictive density for new response y_***

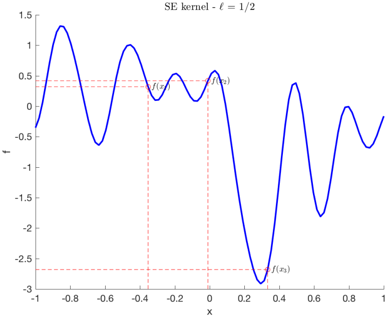
$$\mathrm{y}_*|\mathrm{x}_*, \mathrm{X}, \mathrm{y} \sim N\left(\mathrm{x}_*^T\bar{\mathrm{w}}, \mathrm{x}_*^T\mathrm{A}^{-1}\mathrm{x}_* + \sigma_n^2\right)$$

A prior on w is really a prior over functions



Non-parametric regression

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



- A *new* parameter for *every* x!
- Instead of restricting to linear, impose "prior 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 (thank you, Kolmogorov!)

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?

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

- 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.

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 ($\ell > 0$, $\sigma_f > 0$)

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

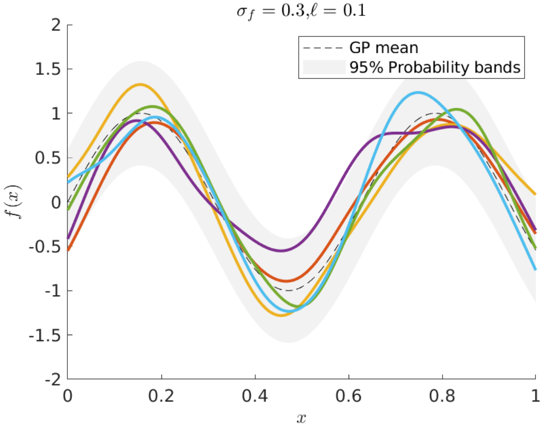
- **Matérn** kernel ($\ell > 0$, $\sigma_f > 0$, $\nu > 0$)

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

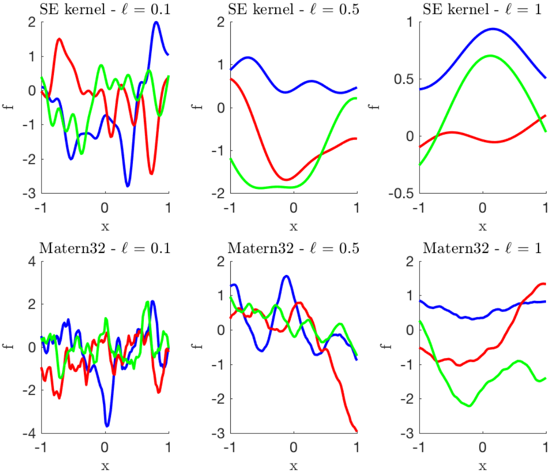
- **Simulate draw** from $f(x) \sim \mathrm{GP}(m(x), k(x, x'))$ by:
  - form a grid $\mathrm{x}_* = (x_1, ..., x_n)$
  - simulate function values from multivariate normal:

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

Simulating a GP

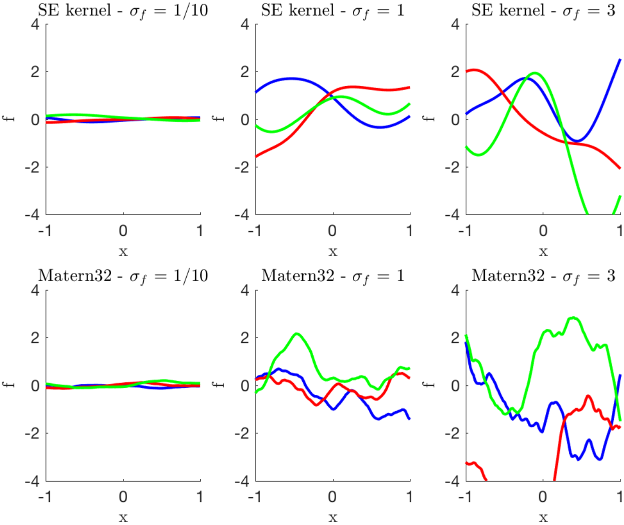


The length scale ℓ - the correlation distance

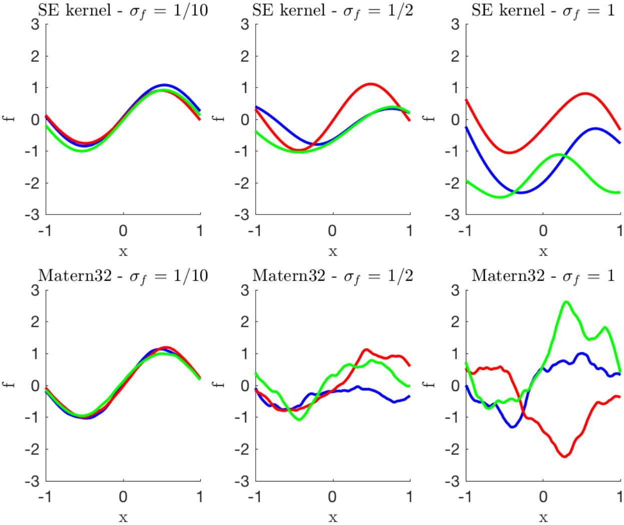


- SE: expected number of zero-crossings on $[0, 1]$: $(2\pi\ell)^{-1}$ (Eq. 4.3)

The scale factor σ_f determines the variance



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



Sequential simulation of GPs

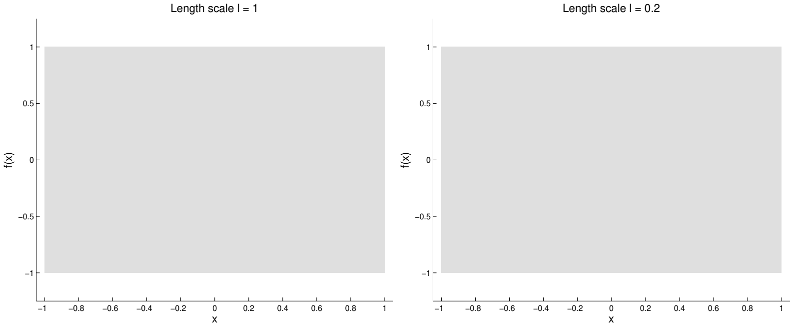
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

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

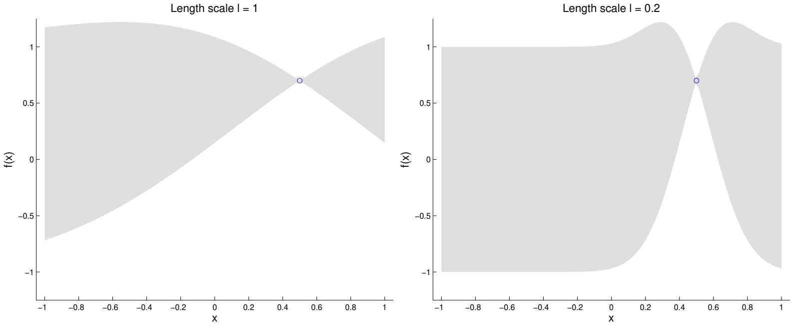
- More intuition from the conditional decomposition

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

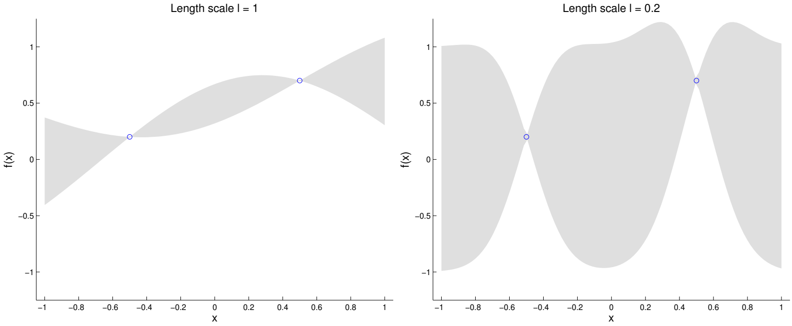
Simulating from $p(f(x_1))$



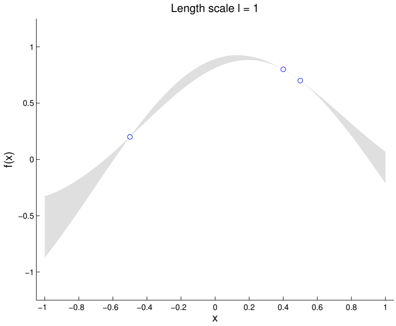
Simulating from $p\left(f\left(x_2\right) \middle| f\left(x_1\right)\right)$

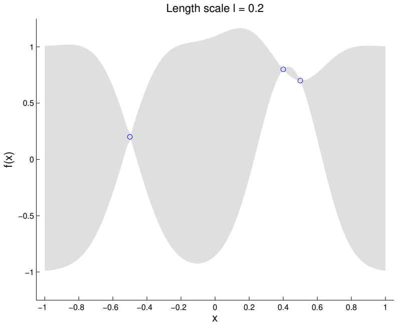


Simulating from $p\left(f\left(x_3\right)\middle|f\left(x_1\right), f\left(x_2\right)\right)$



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





Multivariate normal distribution

- The density of the p-variate $\mathrm{x} \sim N(\mu, \Sigma)$ is

$$f(\mathrm{x}) = \left(\frac{1}{2\pi}\right)^{p/2} \frac{1}{\sqrt{\det \Sigma}} \exp\left\{-\frac{1}{2}(\mathrm{x}-\mu)'\Sigma^{-1}(\mathrm{x}-\mu)\right\}$$

- **Linear combinations**. Let $\mathrm{y} = \mathrm{Bx} + \mathrm{b}$, then

$$\mathrm{y} \sim N(\mathrm{B}\mu + \mathrm{b}, \mathrm{B}\Sigma\mathrm{B}')$$

- Let $\mathrm{x} = \begin{pmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \end{pmatrix}$ where x_1 is $p_1 \times 1$ and x_2 is $p_2 \times 1$ and

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \text{ and } \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

- **Marginals are normal**. Let $\mathrm{x} \sim N(\mu, \Sigma)$, then

$$\mathrm{x}_1 \sim N(\mu_1, \Sigma_{11})$$

- **Conditionals are normal**. Let $\mathrm{x} \sim N(\mu, \Sigma)$, then

$$\mathrm{x}_2 | \mathrm{x}_1 = \mathrm{x}_1^* \sim N\left[\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1^* - \mu_1),\ \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right]$$

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 test data: $\mathrm{f}_* = \mathrm{f}(\mathrm{x}_*)$.
- **Posterior**

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

- **Predictive distribution** for new test data

$$\mathrm{y}_* | \mathrm{x}, \mathrm{y}, \mathrm{x}_* \sim N\left(\bar{\mathrm{f}}_*, \operatorname{cov}(\mathrm{f}_*) + \sigma_n^2 I\right)$$

Sketch 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**

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

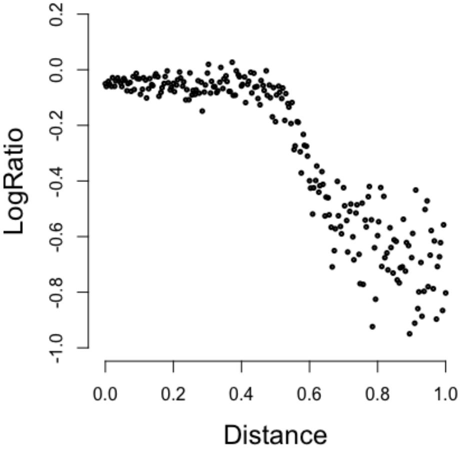
- Joint distribution of $(\mathrm{y}, \mathrm{f}_*)$

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

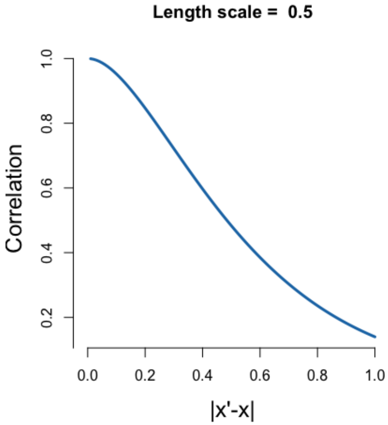
- Complete proof by result:

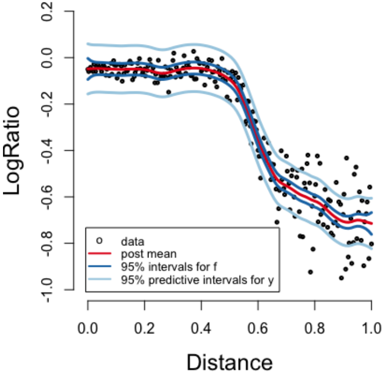
$$\mathrm{x}_2 | \mathrm{x}_1 = \mathrm{x}_1^* \sim N\left[\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1^* - \mu_1),\ \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right]$$

Example - LIDAR data

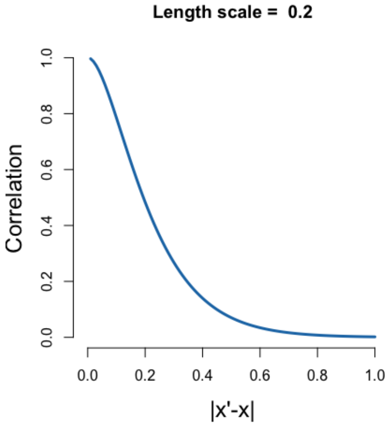


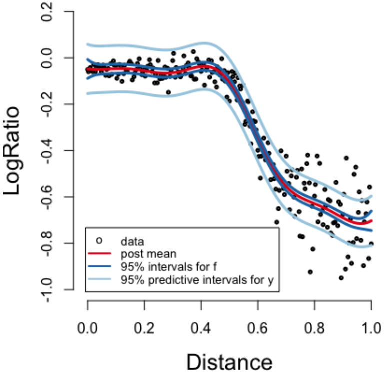
Matern32 $\ell = 0.5, \sigma_f = 0.5, \sigma_n = 0.05$



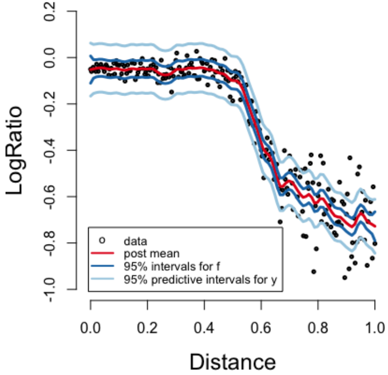


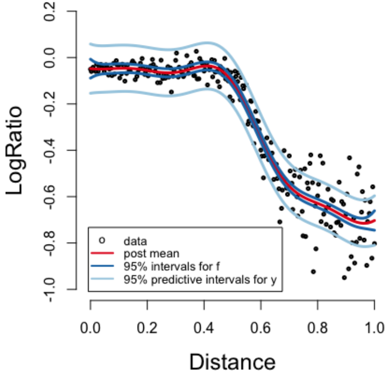
SquaredExp $\ell = 0.2, \sigma_f = 0.5, \sigma_n = 0.05$





Matern32 vs SquaredExp for $\ell = 0.2$

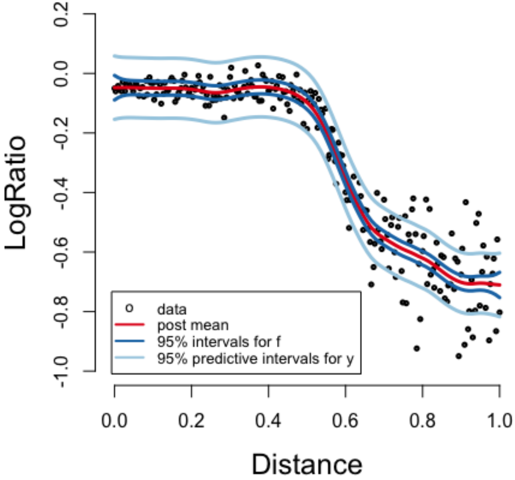




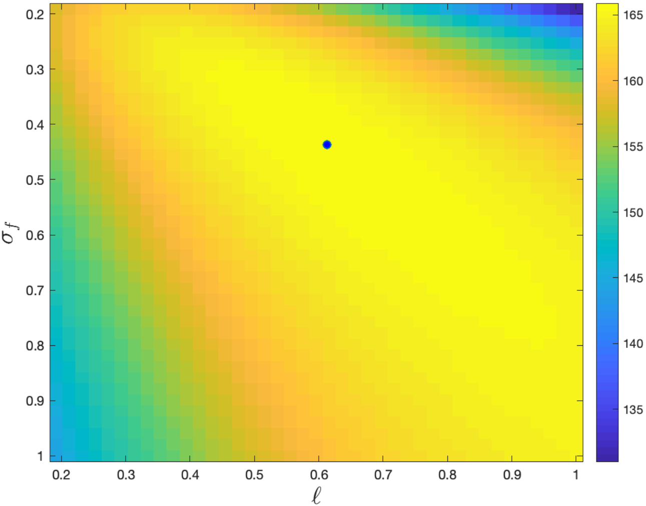
Inference for the hyperparameters

- 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 **GP regression**: $\boldsymbol{y}|\mathrm{X}, \theta \sim N(0, K + \sigma_n^2 I)$ so
$$\log p(\mathrm{y}|\mathrm{X}, \theta) = -\frac{1}{2}\mathrm{y}^T \left(K + \sigma_n^2 I\right)^{-1} \mathrm{y} - \frac{1}{2}\log\left|K + \sigma_n^2 I\right| - \frac{n}{2}\log(2\pi)$$
- Proper **Bayesian inference** **for hyperparameters** (HMC?)
$$p(\theta|\mathrm{y}, \mathrm{X}) \propto p(\mathrm{y}|\mathrm{X}, \theta) p(\theta).$$
- Choice of kernel family by Bayesian model inference. For kernel $K_i \in \mathcal{K}$: $p(K_i|\mathrm{y}, \mathrm{X}) \propto p(\mathrm{y}|\mathrm{X}, K_i) p(K_i)$.

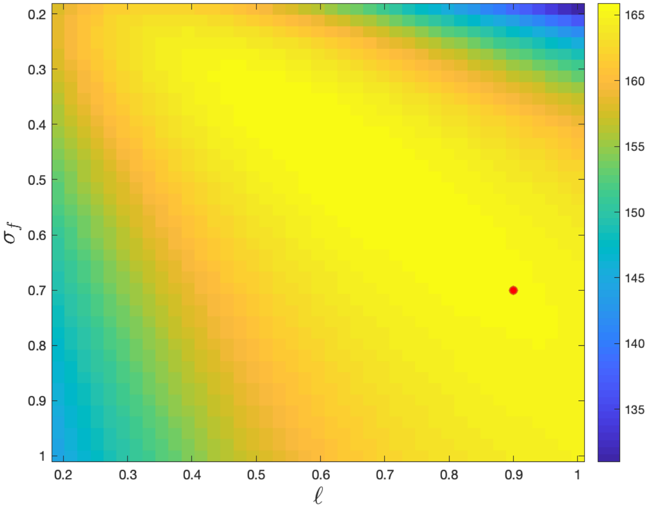
Matern32 $\ell_{opt} = 0.61, \sigma_{f,opt} = 0.44, \sigma_n = 0.05$



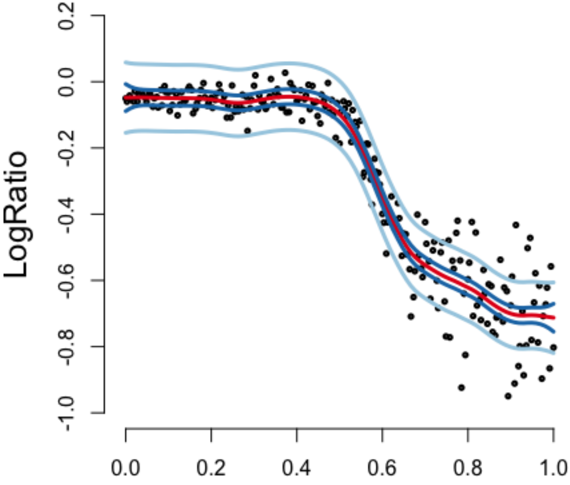
log marginal likelihood surface $\sigma_n = 0.05$



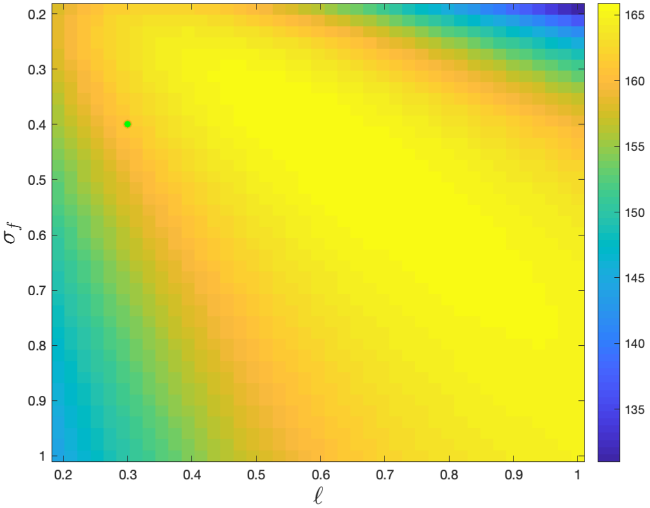
log marginal likelihood surface $\sigma_n = 0.05$



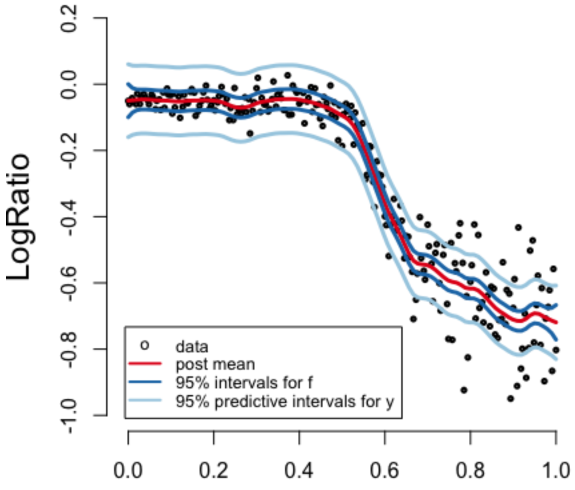
Matern32 $\ell = 0.9, \sigma_f = 0.70, \sigma_n = 0.05$



log marginal likelihood surface $\sigma_n = 0.05$



Matern32 $\ell = 0.3, \sigma_f = 0.4, \sigma_n = 0.05$



GP computations

- Covariance matrix K often numerically singular.
- Noise helps: $K + \sigma_n^2 I$.
- Artificial **jittering** $K + \epsilon I$ for small ϵ.
- Algorithm 2.1 and 3.1 in GPML for stable computations.
- We need to compute:
  - $\left(K + \sigma_n^2 I\right)^{-1} \boldsymbol{y}$ (posterior)
  - $\boldsymbol{y}^T \left(K + \sigma_n^2 I\right)^{-1} \boldsymbol{y}$ (log marginal likelihood)
  - $\left|\log K + \sigma_n^2 I\right|$ (log marginal likelihood)
- $\left(K + \sigma_n^2 I\right)^{-1} \boldsymbol{y}$ corresponds to solving $\left(K + \sigma_n^2 I\right) \boldsymbol{x} = \boldsymbol{y}$ wrt $\boldsymbol{x}$.

Cholesky + Forward and Backward substitution

- Solving $\boldsymbol{Ax} = \boldsymbol{b}$ by $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ is numerically unstable.
- Cholesky factorization $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$ where $\boldsymbol{L}$ is lower triangular.
- Let $\boldsymbol{Ax} = \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{Lz} = \boldsymbol{b}$ if we define $\boldsymbol{z} = \boldsymbol{L}^T\boldsymbol{x}$.
- Solve $\boldsymbol{Lz} = \boldsymbol{b}$ wrt $\boldsymbol{z}$ by forward substitution: $\boldsymbol{z} = \boldsymbol{L}\backslash\boldsymbol{b}$
- Solve $\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{z}$ wrt $\boldsymbol{x}$ by backward substitution: $\boldsymbol{x} = \boldsymbol{L}^T\backslash\boldsymbol{z}$.
- So: $\boldsymbol{x} = \boldsymbol{L}^T\backslash(\boldsymbol{L}\backslash\boldsymbol{b})$.
- $|\boldsymbol{A}| = \left|\boldsymbol{L}\boldsymbol{L}^T\right| = |\boldsymbol{L}|^2 = \left(\prod_{i=1}^p L_{ii}\right)^2$.
- $\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} = \boldsymbol{y}^T(\boldsymbol{L}\boldsymbol{L}^T)^{-1}\boldsymbol{y} = (\boldsymbol{L}\backslash\boldsymbol{y})^T(\boldsymbol{L}\backslash\boldsymbol{y})$.
- Cholesky also preserves sparsity.[1]
- Pre-conditioned conjugate gradient (PCG). $\boldsymbol{Ax} \approx \boldsymbol{b}$. Fast.

[1] Rue and Held (2005). Gaussian Markov random fields. C&H.