Advanced Machine Learning
Gaussian processes
Lecture 1

Mattias Villani

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

Topic overview

- Lecture 1
  - Recall: **The multivariate normal distribution**
  - Recall: Bayesian inference for **Gaussian linear/nonlinear regression**
  - Introduction to **Gaussian Process Regression**
- Lecture 2
  - **More on kernel functions**
  - Estimating the **GP hyperparameters**
  - **Large scale GPs**
- Lecture 3
  - **Gaussian Process Classification**
  - Some examples of **GP applications**

The multivariate normal distribution

- The **density function** of a p-variate normal vector $\mathbf{x} \sim N(\mu, \Sigma)$ is

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

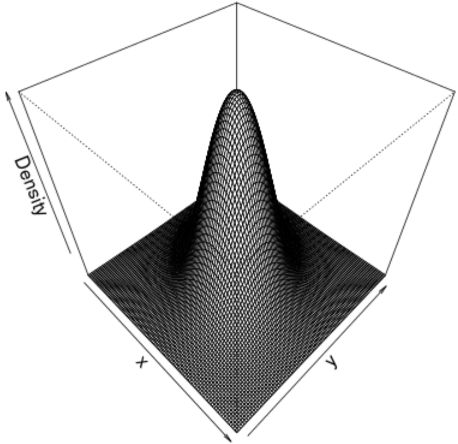
- Example: **Bivariate normal** $(p = 2)$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

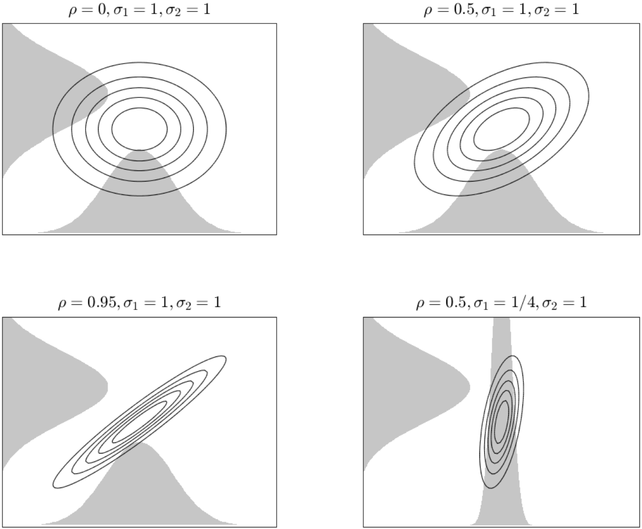
- Mean and variance

$$E(\mathbf{x}) = \mu \quad Var(\mathbf{x}) = \Sigma$$

Multivariate normal



Multivariate normal



The multivariate normal distribution, cont.

- Let $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}$ where $\mathbf{x}_1$ is $p_1 \times 1$ and $\mathbf{x}_2$ is $p_2 \times 1$ $(p_1 + p_2 = p)$.
- Partition μ and Σ accordingly as

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

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

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

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

$$\mathbf{x}_1 | \mathbf{x}_2 = \mathbf{x}_2^* \sim N\left[\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2^* - \mu_2),\ \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right]$$

Nonlinear regression

- **Linear regression**[1]
$$y = f(\mathbf{x}) + \epsilon$$
$$f(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x}$$
and $\epsilon \sim N(0, \sigma_n^2)$ and iid over observations.
- The weights $\mathbf{w}$ are called regression coefficients (β) in statistics.
- **Polynomial regression**: $\phi(\mathbf{x}) = (1, x, x^2, x^3, ..., x^k)$:
$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \cdot$$
- More generally: **splines** with **basis functions**.
- Polynomial and spline models are linear in $\mathbf{w}$. Least squares!

[1] I follow the notation in RW rather than PRML. In PRML: y is the noise-free response. $t = y + \epsilon$ is the response with noise. $\beta^{-\mathbf{1}}$ is the noise variance ($\sigma_n^{\mathbf{2}}$).

Bayesian linear regression - inference

- Linear regression for all n observations

$$\underset{n\times 1}{\mathbf{y}} = \underset{n\times p}{\mathbf{X}} \underset{p\times 1}{\mathbf{w}} + \underset{n\times 1}{\varepsilon}$$

- $\mathbf{w}$ is unknown. σ_n is assumed known.
- **Prior**

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

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

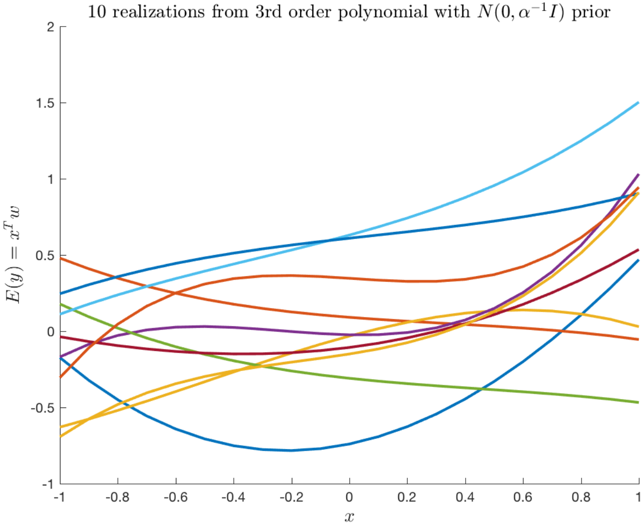
$$\mathbf{w}|\mathbf{X},\mathbf{y} \sim N\left(\bar{\mathbf{w}}, \mathbf{A}^{-1}\right)$$

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

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

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

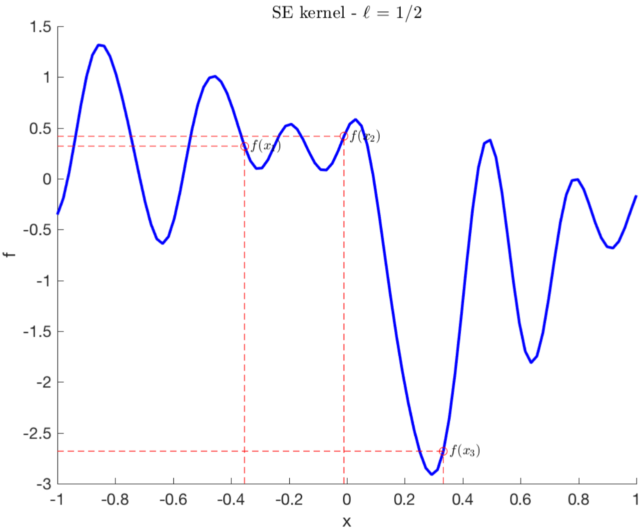
A prior on **w** is a prior on functions



Non-parametric regression

- **Non-parametric regression**: avoiding a parametric form for $f(\cdot)$. Treat $f(\mathbf{x})$ as an unknown parameter for every $\mathbf{x}$.
- **Weight space view**
  - Restrict attention to a grid of (ordered) x-values: $x_1, x_2, .., x_k$.
  - Put a joint prior on the k function values: $f(x_1), f(x_2), ..., f(x_k)$.
- **Function space view**
  - Treat f as an **unknown function**.
  - Put a **prior over a set of functions**.

Nonparametric = one parameter for every x!



Gaussian process regression

- Weight-space view. GP assumes

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

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

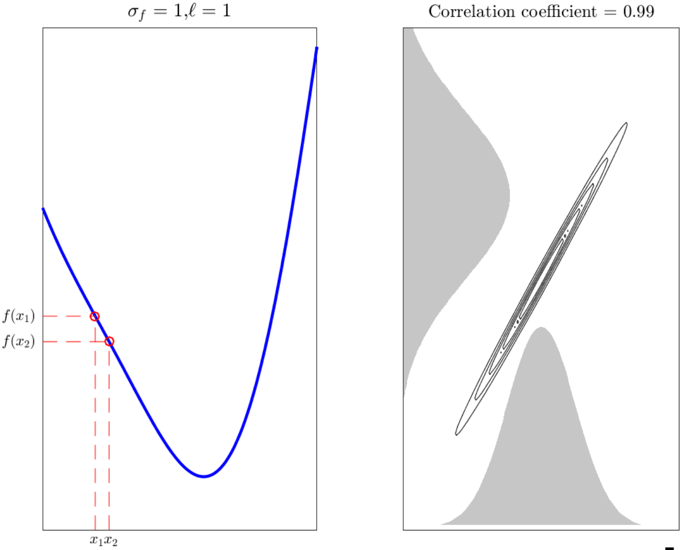
$$Cov\left(f(x_p), f(x_q)\right)$$

- **Squared exponential covariance function**

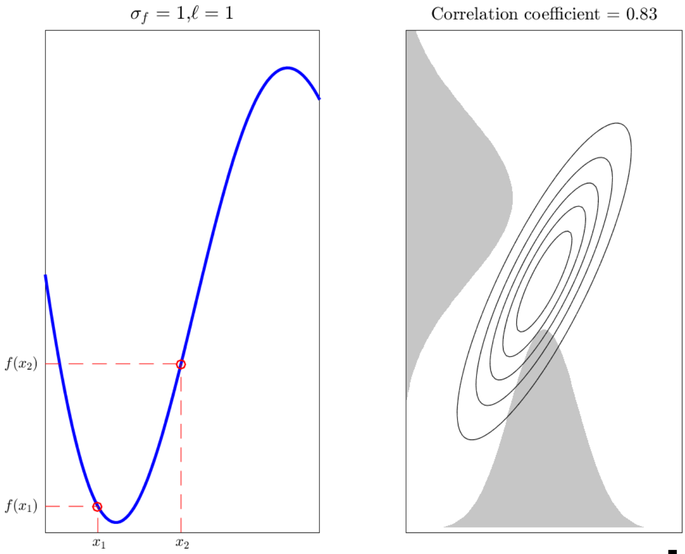
$$Cov\left(f(x_p), f(x_q)\right) = k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2}\left(x_p - x_q\right)^2\right)$$

- 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.
- Extension to multiple covariates: $(x_p - x_q)^2$ replaced by $(\mathbf{x}_p - \mathbf{x}_q)^T(\mathbf{x}_p - \mathbf{x}_q)$.

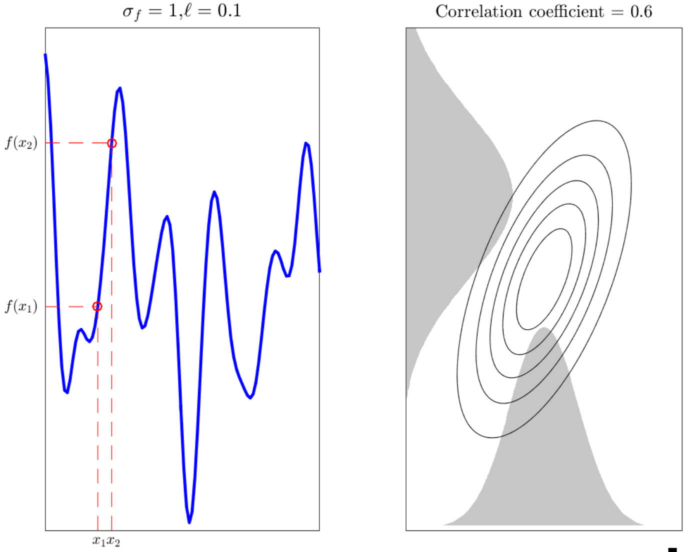
SMOOTH FUNCTION - POINTS NEARBY



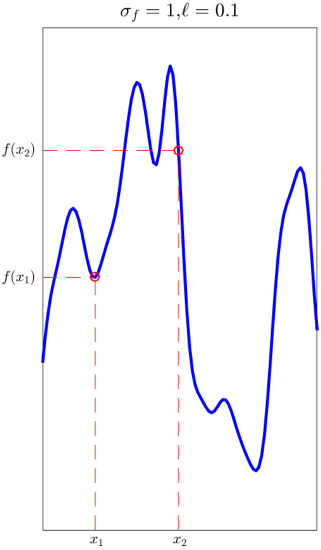
Smooth function - points far apart

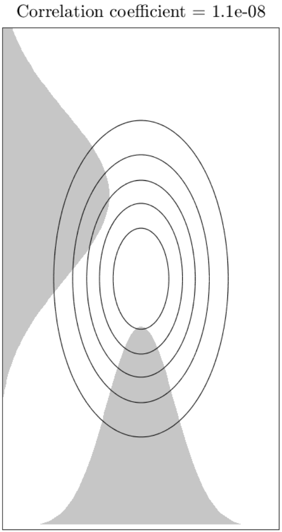


JAGGED FUNCTION - POINTS NEARBY



Jagged function - points far apart





Gaussian process regression, cont.

Definition

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

- A Gaussian process is really a **probability distribution** **over functions** (curves).
- A GP is completely 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' (note: this is *not* the transpose here).
- A **Gaussian process** is denoted by

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

- **Bayesian**: $f(x) \sim GP$ encodes **prior beliefs** about the unknown $f(\cdot)$.

Simulating a GP

- Example:

$$m(x) = \sin(10x)$$
$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{x - x'}{\ell}\right)^2\right)$$

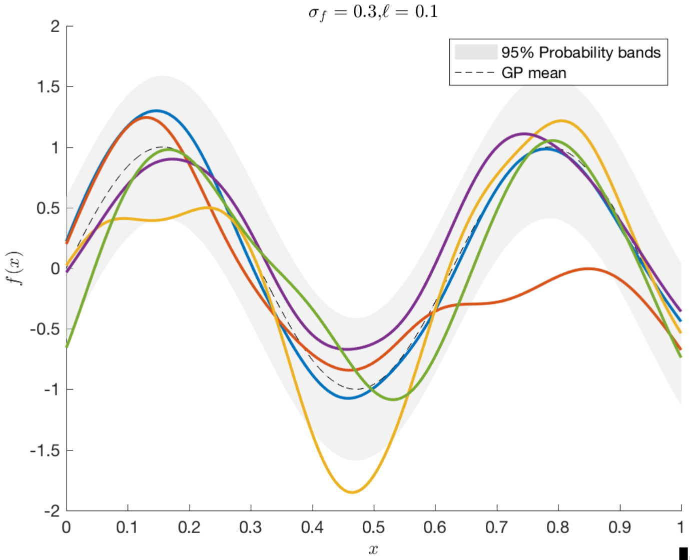
  where $\ell > 0$ is the length scale.

- Larger ℓ gives more smoothness in $f(x)$.
- Simulate draw from $f(x) \sim GP\left(m(x), k(x, x')\right)$ over a grid $\mathbf{x}_* = (x_1, ..., x_n)$ by using that

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

- Note that the **kernel** $k(x, x')$ produces a **covariance matrix** $K(\mathbf{x}_*, \mathbf{x}_*)$ when evaluated at the vector $\mathbf{x}_*$.

Simulating a GP



Three commonly used covariance kernels

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

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

  - Infinitely mean square differentiable. Very smooth.
- **Rational Quadratic (RQ)** ($\ell > 0$, $\sigma_f > 0$,$\alpha > 0$)

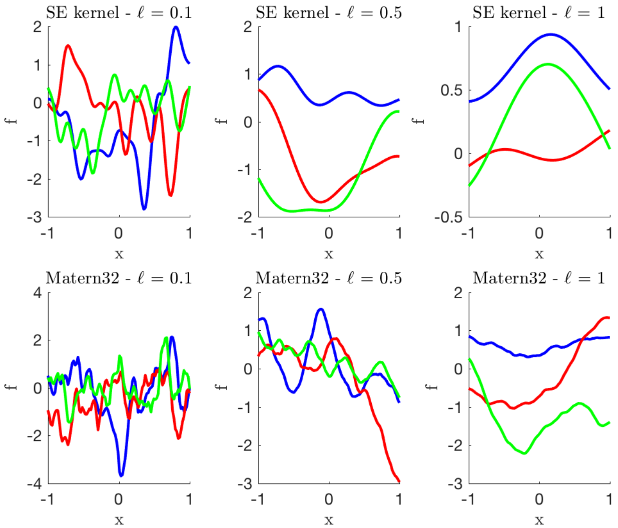
$$K_{RQ}(r) = \sigma_f \left(1 + \frac{r^2}{2\alpha\ell^2}\right)^{-\alpha}$$

  - RQ is sum of SE with different ℓ. When $\alpha \to \infty$, $K_{RQ}(r) \to K_{SE}(r)$.
- **Matérn** ($\ell > 0$, $\sigma_f > 0$, $\nu > 0$)

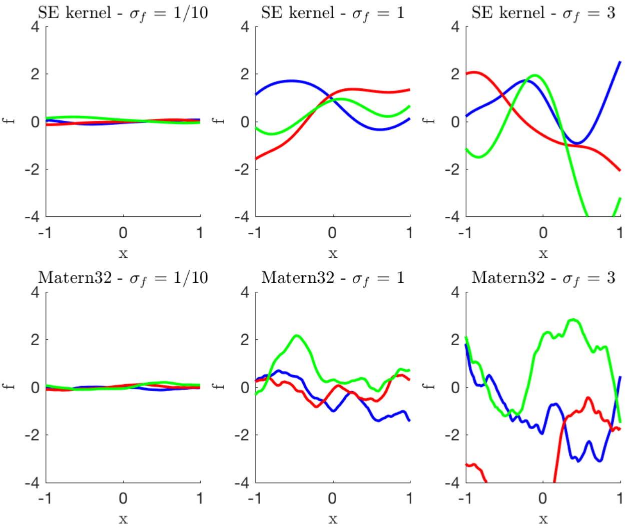
$$K_{Matern}(r) = \sigma_f \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r}{\ell}\right)$$

  - $\nu = 3/2$ and $\nu = 5/2$ common. As $\nu \to \infty$, $K_{Matern}(r) \to K_{SE}(r)$.

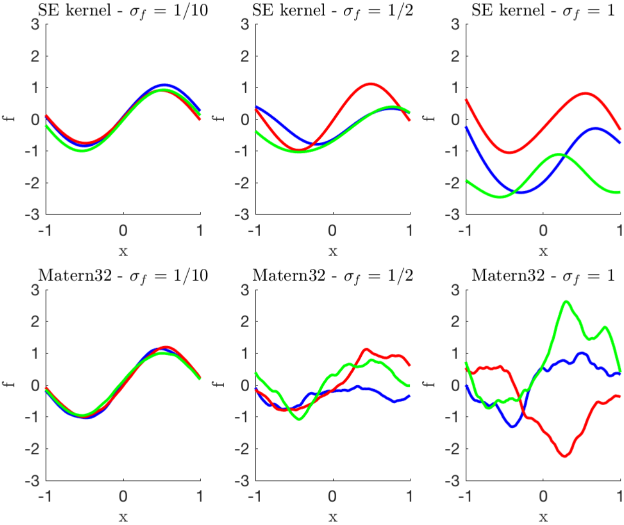
THE LENGTH SCALE ℓ DETERMINES THE SMOOTHNESS



The scale factor σ_f determines the variance



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



Simulating a GP

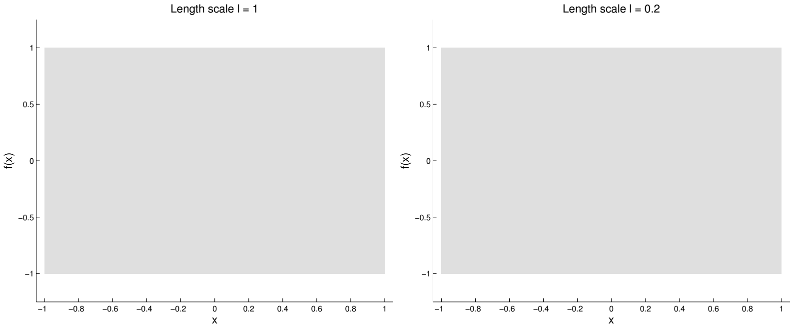
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

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

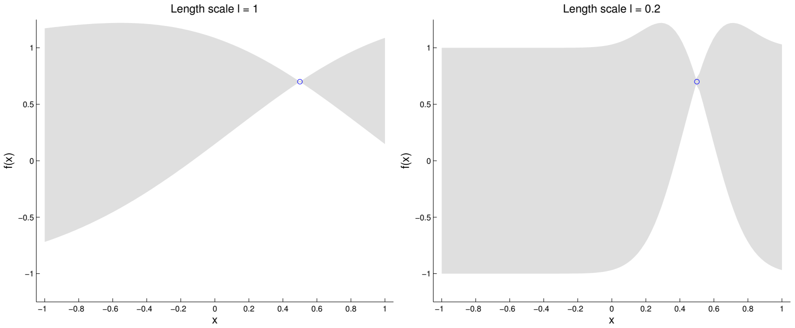
- More intuition from the conditional decomposition

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

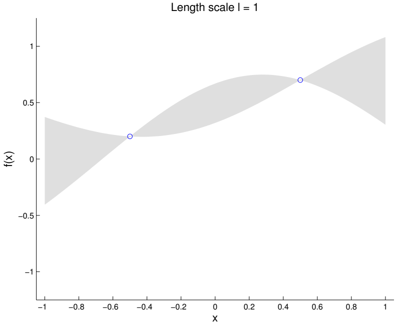
SIMULATION FROM ℓ=1 VS ℓ=0.2. BEFORE FIRST DRAW.



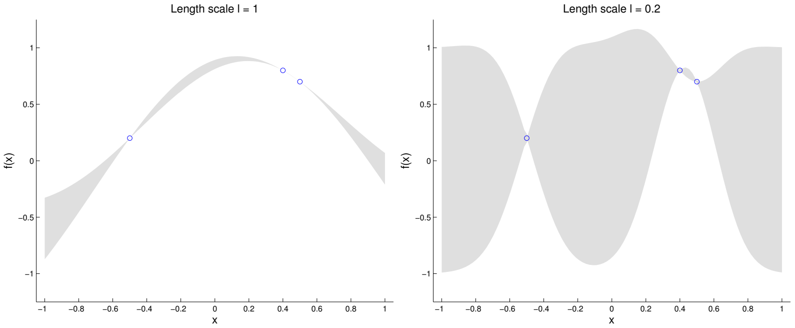
Simulation from ℓ=1 vs ℓ=0.2. Before second draw.



Simulation from ℓ=1 vs ℓ=0.2. Before third draw.



Simulation from ℓ=1 vs ℓ=0.2. Before fourth draw.



The posterior for a Gaussian Process Regression

- **Model**

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

- **Prior**

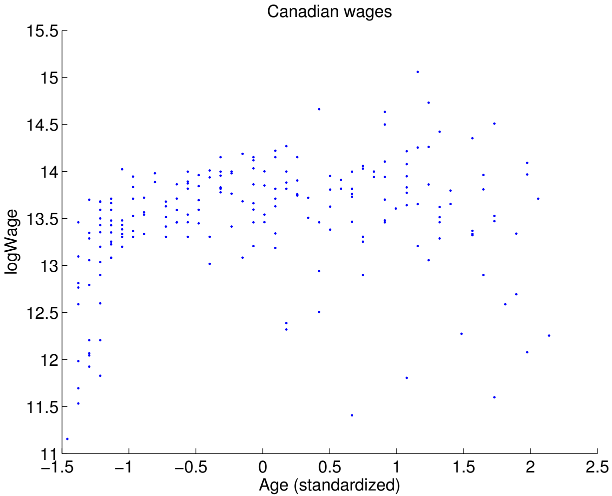
$$f(x) \sim GP\left(0, k(x, x')\right)$$

- You have observed the data: $\mathbf{x} = (x_1, ..., x_n)^T$ and $\mathbf{y} = (y_1, ..., y_n)^T$.
- Goal: the posterior of $f(\cdot)$ over a grid of x-values: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.
- The **posterior** (use formula for conditional Gaussian above)

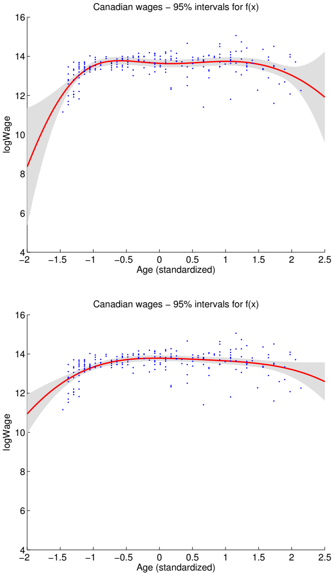
$$\mathbf{f}_* | \mathbf{x}, \mathbf{y}, \mathbf{x}_* \sim N\left(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)\right)$$

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

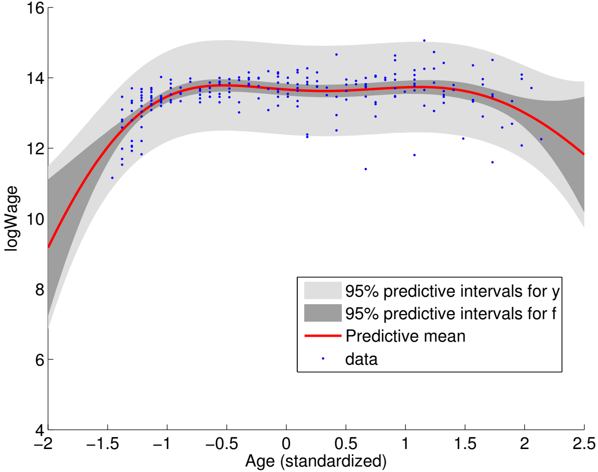
Example - Canadian wages



Posterior of f - $\ell = 0.2, 0.5, 1, 2$



Canadian wages - prediction with $\ell = 0.5$



Software

- Python: GPy
- Matlab: Statistics and Machine Learning Toolbox, GPML, GPstuff.
- R: Kernlab,