Introduction to Machine Learning
Topic 6: Gaussian processes and mixture models
Lecture 1

Mattias Villani

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

Topic overview

- Recall: **The multivariate normal distribution**
- Bayesian inference for **Gaussian linear/nonlinear regression**.
- **Gaussian processes** for **nonparametric regression**
  - Covariance kernels
  - Selecting the kernel and hyperparameters
- **Introduction to GP classification**

The multivariate normal distribution

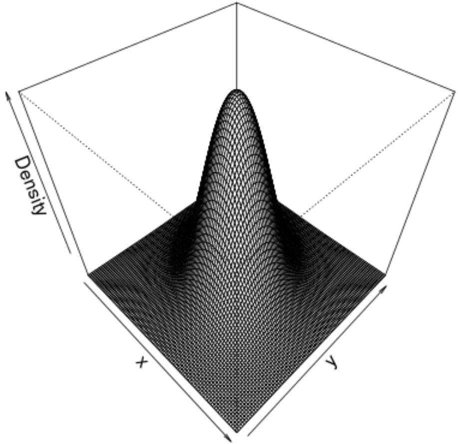
- The **density function** of a p-variate normal vector $\mathbf{x} \sim N(\mu, \Sigma)$ is

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

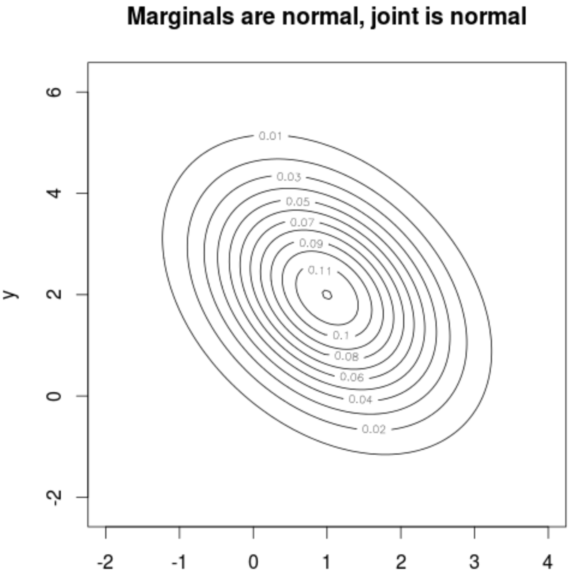
- Example: **Bivariate normal** $(p = 2)$

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

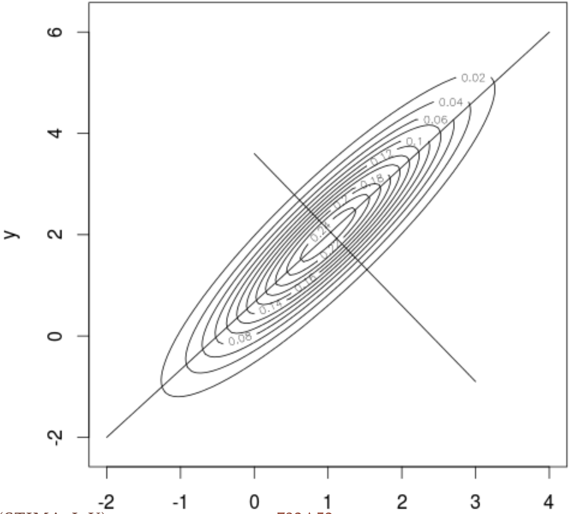
Multivariate normal



Multivariate normal



Multivariate normal



Flexible nonlinear regression

- **Linear regression**

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

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

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

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \cdot \mathbf{w}$$

- More generally: **splines** with **basis functions**. See Topic 7 in this course.

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**

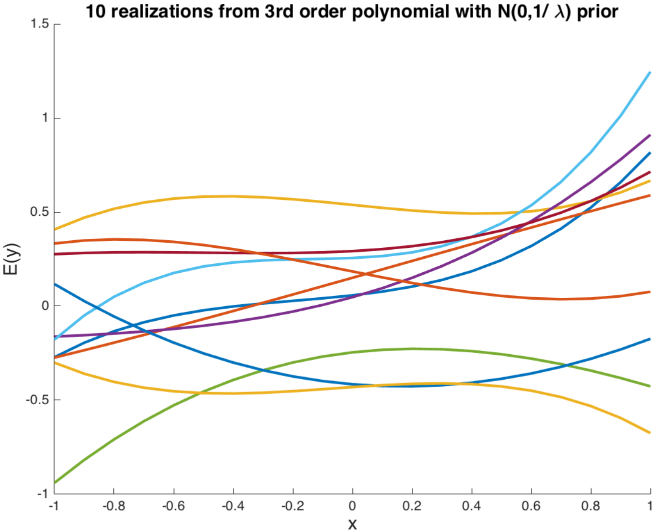
$$\mathbf{w} \sim N\left(0, \Sigma_p\right)$$

- Ridge regression: $\Sigma_p = \lambda^{-1} I$.
- **Posterior**

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

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

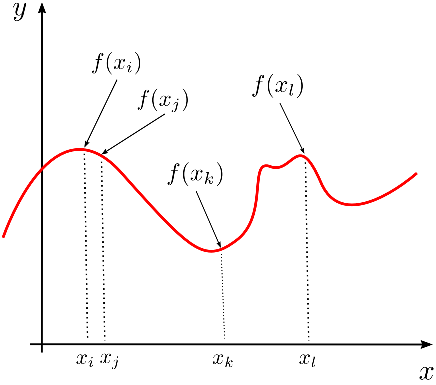
Flexible nonlinear regression



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**?

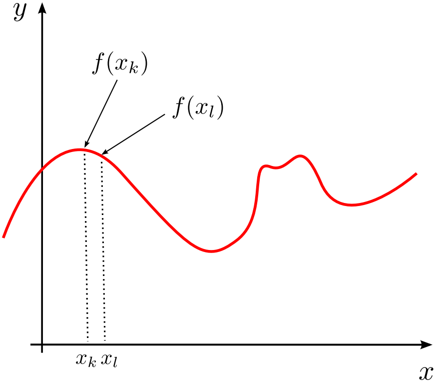
$$Cov\left(f(x_p), f(x_q)\right)$$

- Squared exponential **covariance function**

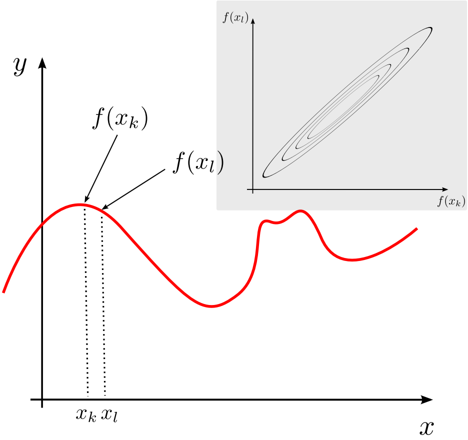
$$Cov\left(f(x_p), f(x_q)\right) = k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2}\left(x_p - x_q\right)^2\right)$$

- 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: $\left(x_p - x_q\right)^2$ replaced by $(\mathbf{x}_p - \mathbf{x}_q')^T(\mathbf{x}_p - \mathbf{x}_q')$.

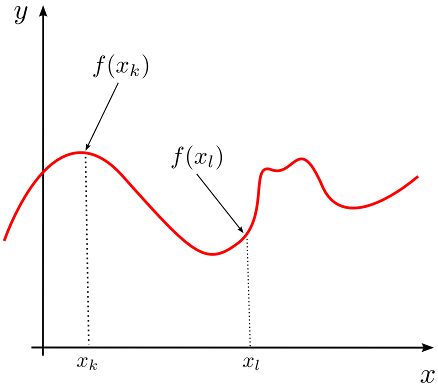
Smooth function - points nearby



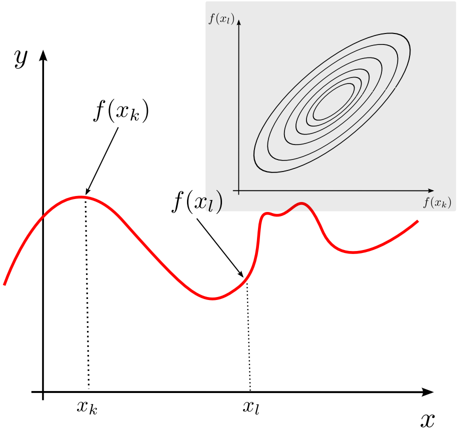
SMOOTH FUNCTION - POINTS NEARBY



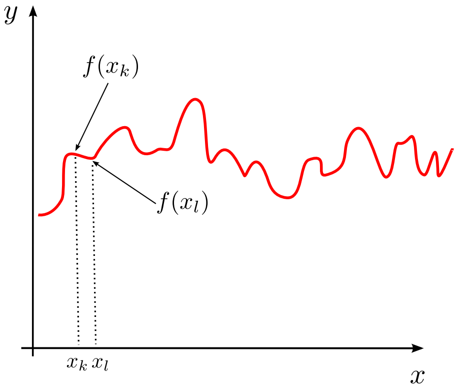
Smooth function - points far apart



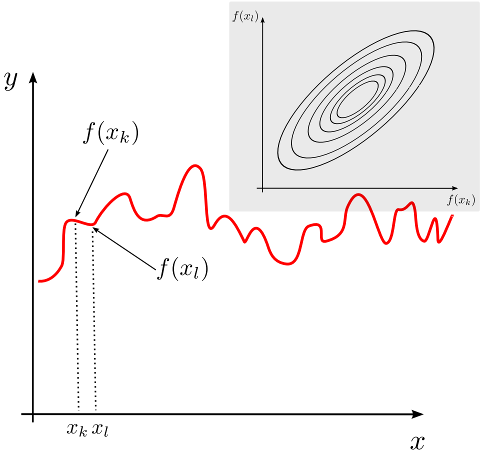
Smooth function - points far apart



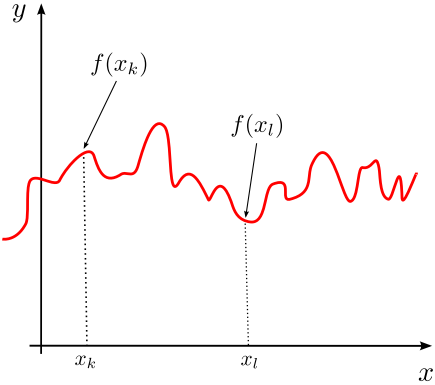
Jagged function - points nearby



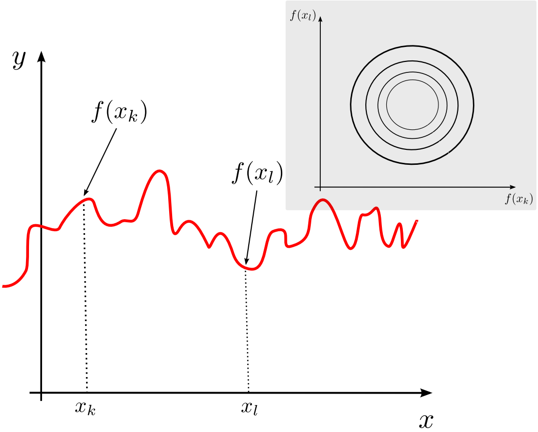
JAGGED FUNCTION - POINTS NEARBY



Jagged function - points far apart



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 $f(\cdot)$.

A SIMPLE GP EXAMPLE

- Example:

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

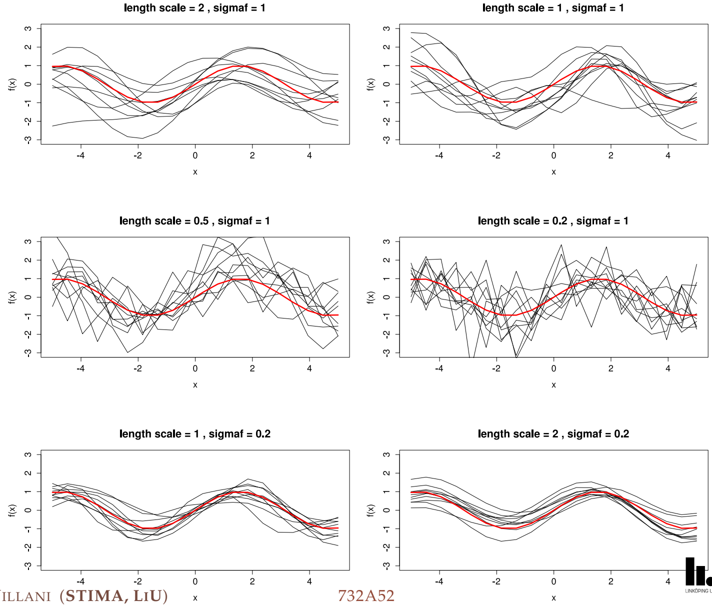
  where $\ell > 0$ is the length scale.

- 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 - sine mean and SE kernel



Simulating a GP

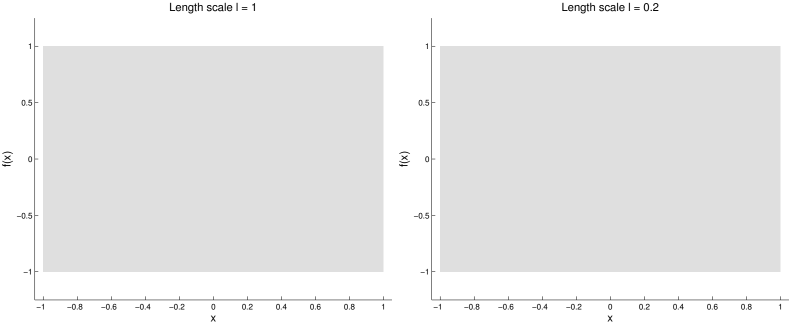
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

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

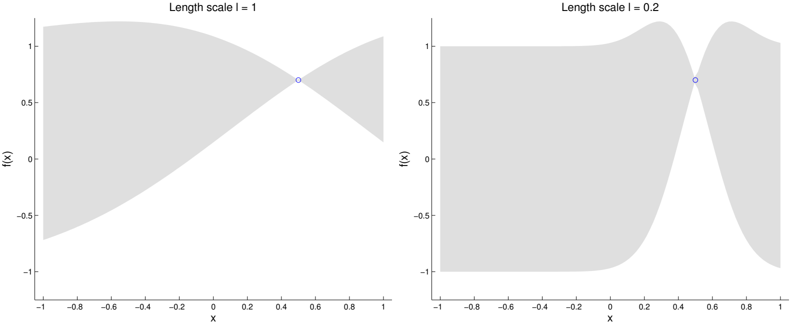
- More intuition from the conditional decomposition

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

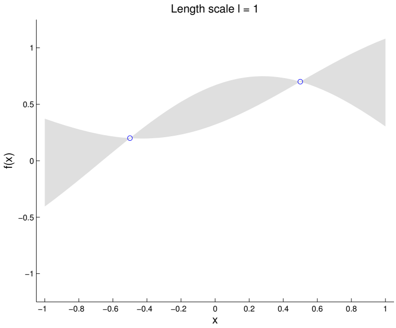
SIMULATION FROM ℓ=1 VS ℓ=0.2. BEFORE FIRST DRAW.



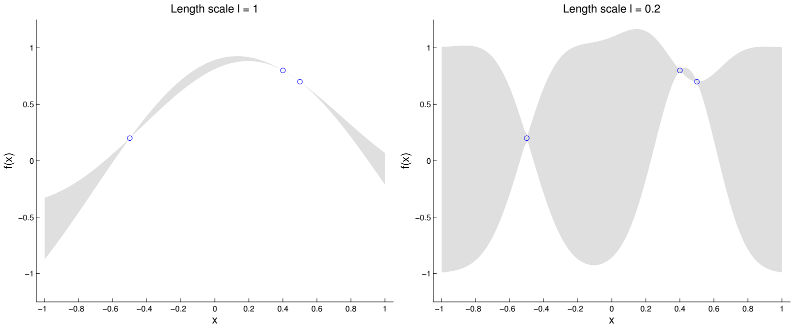
Simulation from ℓ=1 vs ℓ=0.2. Before second draw.



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**

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

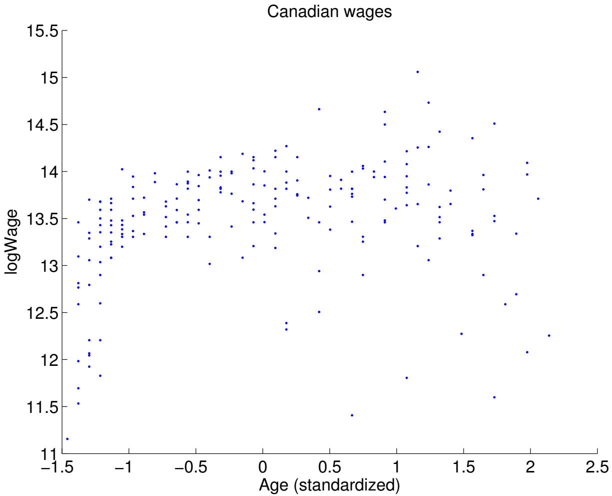
- You have observed the data: $\mathbf{x} = (x_1, ..., x_n)'$ and $\mathbf{y} = (y_1, ..., y_n)'$.
- Goal: the posterior of $f(\cdot)$ over a grid of x-values: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.
- The **posterior** (use formula for conditional Gaussian above)

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

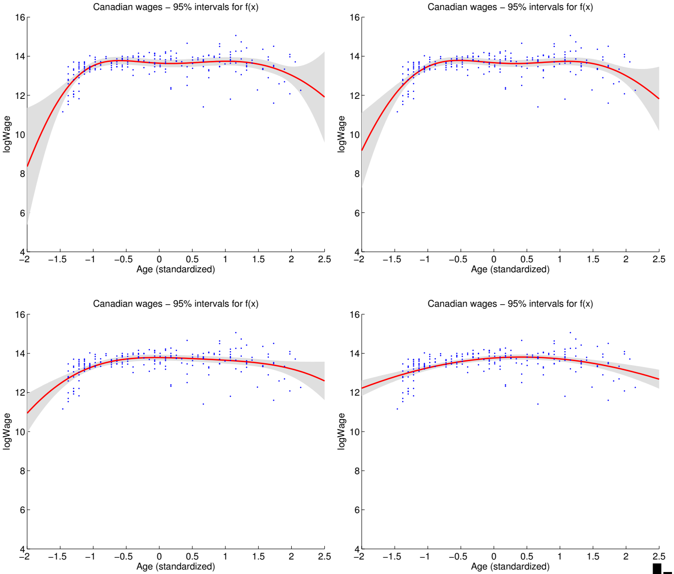
$$\bar{\mathbf{f}}_* = K(\mathbf{x}_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} \mathbf{y}$$

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

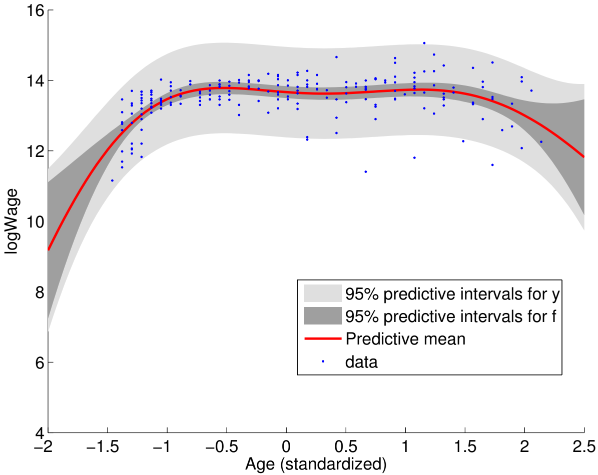
Example - Canadian wages



POSTERIOR OF F - $\ell = 0.2, 0.5, 1, 2$



Canadian wages - prediction with $\ell = 0.5$



Two 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.
- **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 $\nu = 5/2$ most useful for ML. As $\nu \to \infty$, Matérn's kernel approaches SE kernel.

MORE THAN ONE INPUT - ARD

- Anisotropic version of isotropic kernels by setting $r^2(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T \mathbf{M}(\mathbf{x} - \mathbf{x}')$ where $\mathbf{M}$ is positive definite.
- **Automatic Relevance Determination** (**ARD**): $\mathbf{M} = Diag(\ell_1^{-2}, ..., \ell_D^{-2})$ is diagonal with different length scales.
- ARD does 'variable selection' since large ℓ_j means that the jth input essentially drops out of $f(\mathbf{x})$.

Determining the hyperparameters

- Kernel depends on **hyperparameters** θ. Example SE kernel $[\theta = (\sigma_f, \ell)^T]$

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

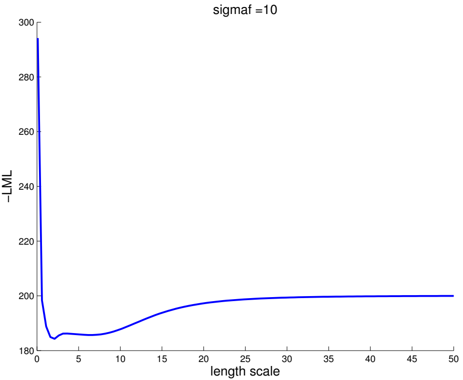
- Common approach: choose the hyperparameters that maximizes the **marginal likelihood** (**evidence**):

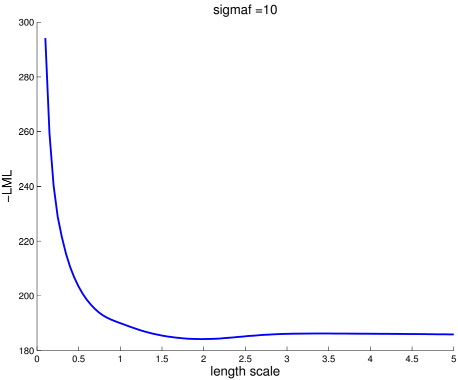
$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{X}, \mathbf{f}, \theta) p(\mathbf{f}|\mathbf{X}, \theta) d\mathbf{f}$$

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

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

Canadian wages - LML determination of ℓ





Classification with logistic regression

- **Classification**: **binary response** $y \in \{-1, 1\}$ predicted by features $\mathbf{x}$.

- Example: linear logistic regression

$$Pr(y = 1|\mathbf{x}) = \lambda(\mathbf{x}^T\mathbf{w})$$

where $\lambda(z)$ is the logistic **link function**

$$\lambda(z) = \frac{1}{1 + \exp(-z)}$$

- $\lambda(z)$ 'squashes' the linear prediction $\mathbf{x}^T\mathbf{w} \in \mathbb{R}$ into $\lambda(\mathbf{x}^T\mathbf{w}) \in [0, 1]$.
- Logistic regression has **linear decision boundaries**.

GP CLASSIFICATION

- Obvious **GP extension** of logistic regression: replace $\mathbf{x}^T\mathbf{w}$ by $f(\mathbf{x})$ where

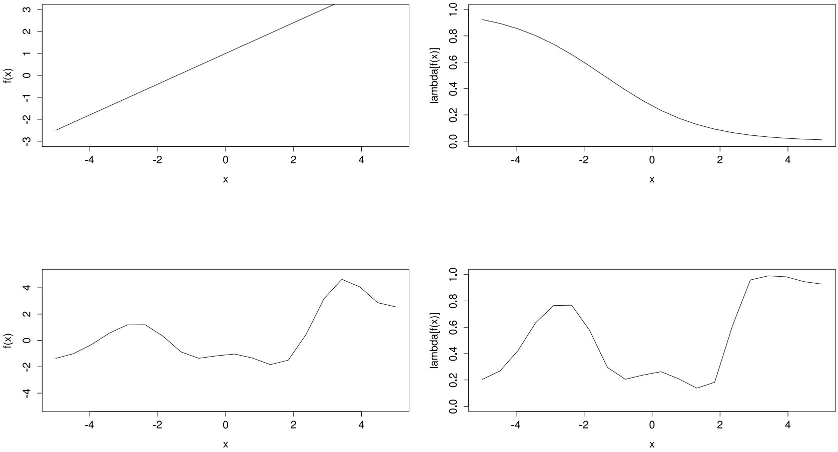
$$f(\mathbf{x}) \sim GP(0, k(\mathbf{x}, \mathbf{x}'))$$

  and squash f through logistic function (or normal CDF)

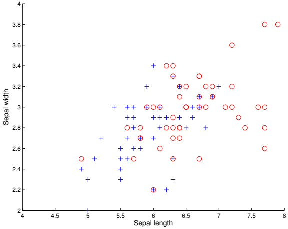
$$Pr(y = 1|\mathbf{x}) = \lambda(f(\mathbf{x}))$$

- Posterior and predictive distribution is complicated. Solutions:
  - Approximations: **Laplace**, **Expectation Propagation** (**EP**) or Variational Bayes (VB)
  - MCMC sampling.

SQUASHING F

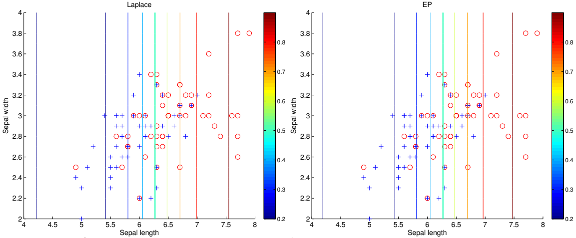


Iris data - sepal - SE kernel with ARD

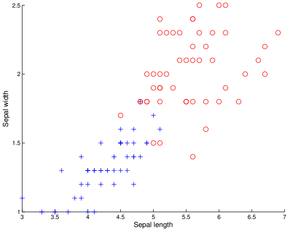


Laplace: $\hat{\ell}_1 = 1.7214, \hat{\ell}_2 = 185.5040, \sigma_f = 1.4361$

EP: $\hat{\ell}_1 = 1.7189, \hat{\ell}_2 = 55.5003, \sigma_f = 1.4343$

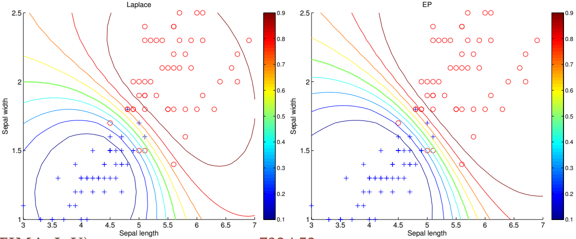


Iris data - petal - SE kernel with ARD

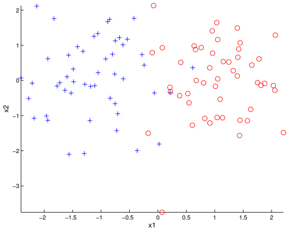


Laplace: $\hat{\ell}_1 = 1.7606, \hat{\ell}_2 = 0.8804, \sigma_f = 4.9129$

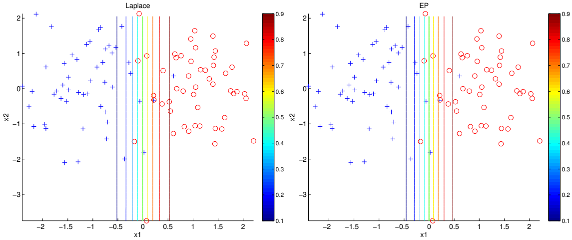
EP: $\hat{\ell}_1 = 2.1139, \hat{\ell}_2 = 1.0720, \sigma_f = 5.3369$



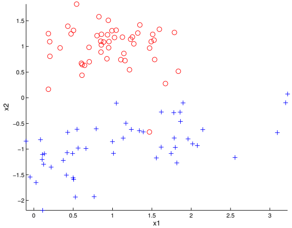
Toy data 1 - SE kernel with ARD



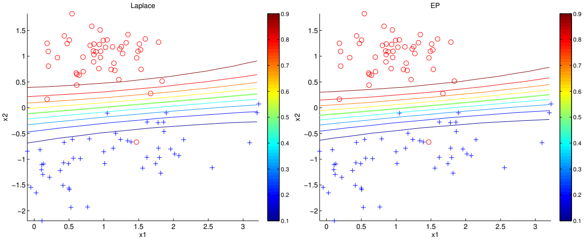
EP: $\hat{\ell}_1 = 2.4503, \hat{\ell}_2 = 721.7405, \sigma_f = 4.7540$



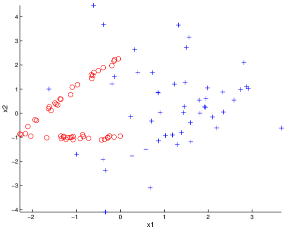
Toy data 2 - SE kernel with ARD



EP: $\hat{\ell}_1 = 8.3831, \hat{\ell}_2 = 1.9587, \sigma_f = 4.5483$



Toy data 3 - SE kernel with ARD



Laplace: $\hat{\ell}_1 = 0.7726, \hat{\ell}_2 = 0.6974, \sigma_f = 11.7854$

EP:$\hat{\ell}_1 = 1.2685, \hat{\ell}_2 = 1.0941, \sigma_f = 17.2774$