Gaussian Processes and Optimization

Mattias Villani

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

Lecture Overview

- Gaussian Process Regression
- Gaussian Process Classification
- Gaussian Process Optimization

Flexible nonlinear 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**: $\mathbf{x} = (1, x, x^2, x^3, ..., x^k)^T$.
- **Spline regression**:

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

  where $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_N(\mathbf{x}))^T$ for N basis functions.
- Example: **thin plate splines** with N knots $\kappa_1, ..., \kappa_N$ in $\mathbf{x}$-space

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

Bayesian linear regression - inference

- β is unknown. σ_n is assumed known.
- **Prior**

$$\beta \sim N(0, \Sigma_p)$$

- **Posterior**

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

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

Bayesian linear regression - prediction

- **Predictive density for mean** $f(\mathbf{x}_*)$ at new location $\mathbf{x}_*$

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

- Proof: $f(\mathbf{x}_*) = \mathbf{x}_*^T\beta$ and β has a normal posterior. Linear combs of normals is normal.

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

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

- Replace $\mathbf{X}$ with $\Phi(\mathbf{X})$ in the above for the case with basis expansion (e.g. splines).

Non-parametric regression

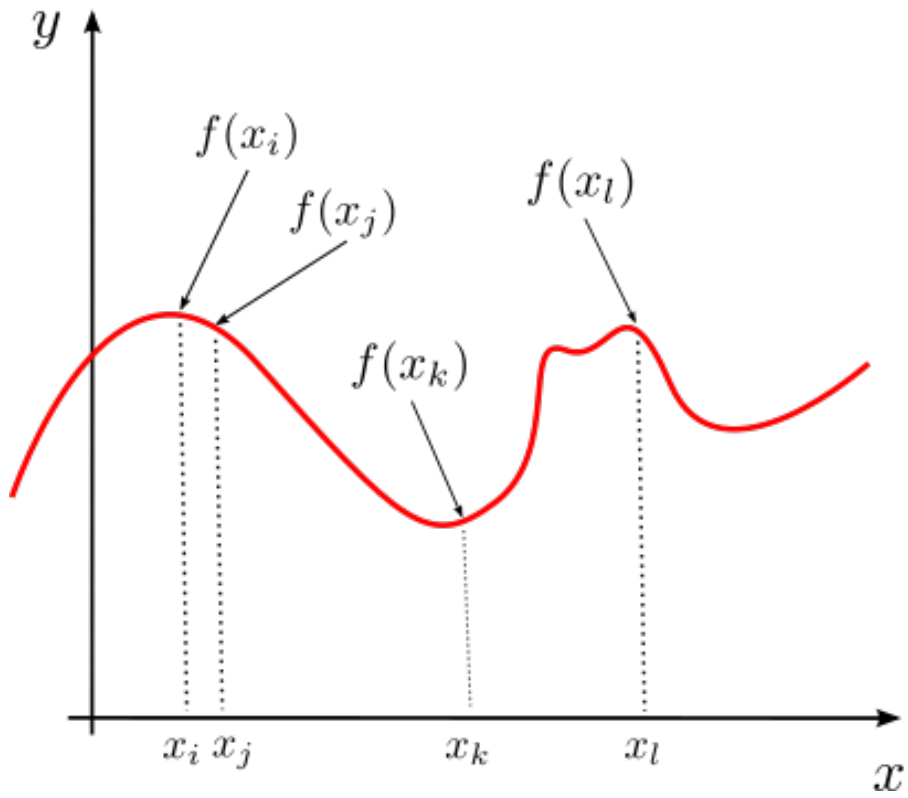
- **Non-parametric regression**: avoiding a parametric form for $f(\cdot)$. Treat $f(\mathbf{x})$ as an unknown parameter for every $\mathbf{x}$.

- **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**.

- The two views are identical for Gaussian processes.

NONPARAMETRIC = ONE PARAMETER FOR EVERY X!



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** $(p = 2)$

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

- **Linear combinations**. Let $\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{b}$, where $\mathbf{x}$ is $p \times 1$ and $\mathbf{B}$ is a $m \times p$ constant matrix. Then

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

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

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

- Life is beautiful ...

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**?

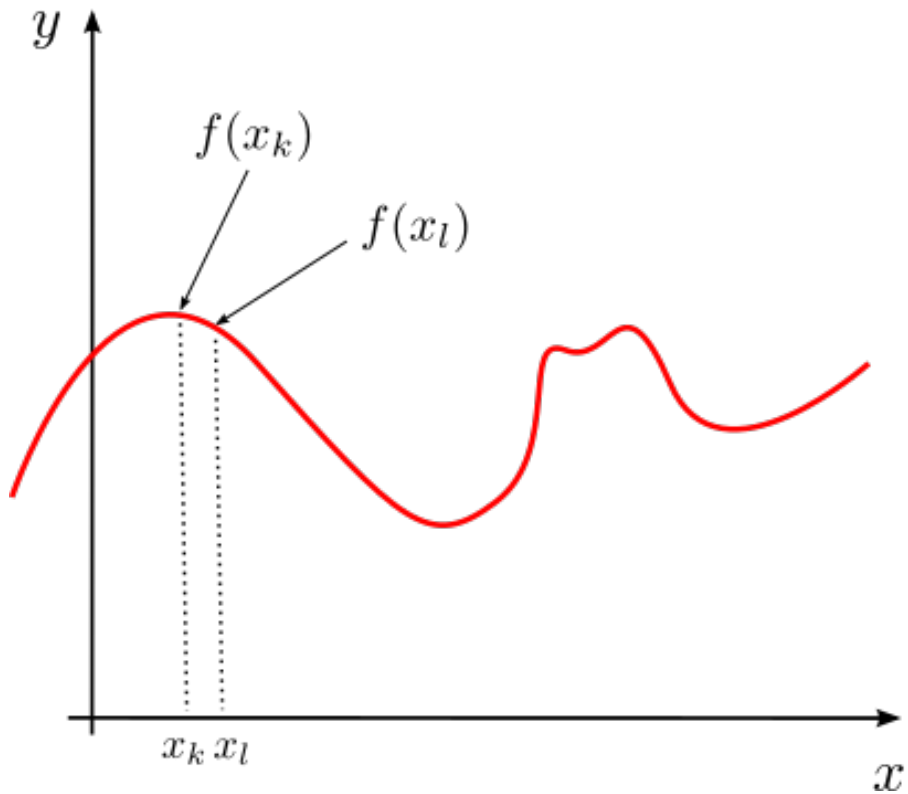
$$Cov\left(f(x_p), f(x_q)\right)$$

- **Squared exponential covariance function**

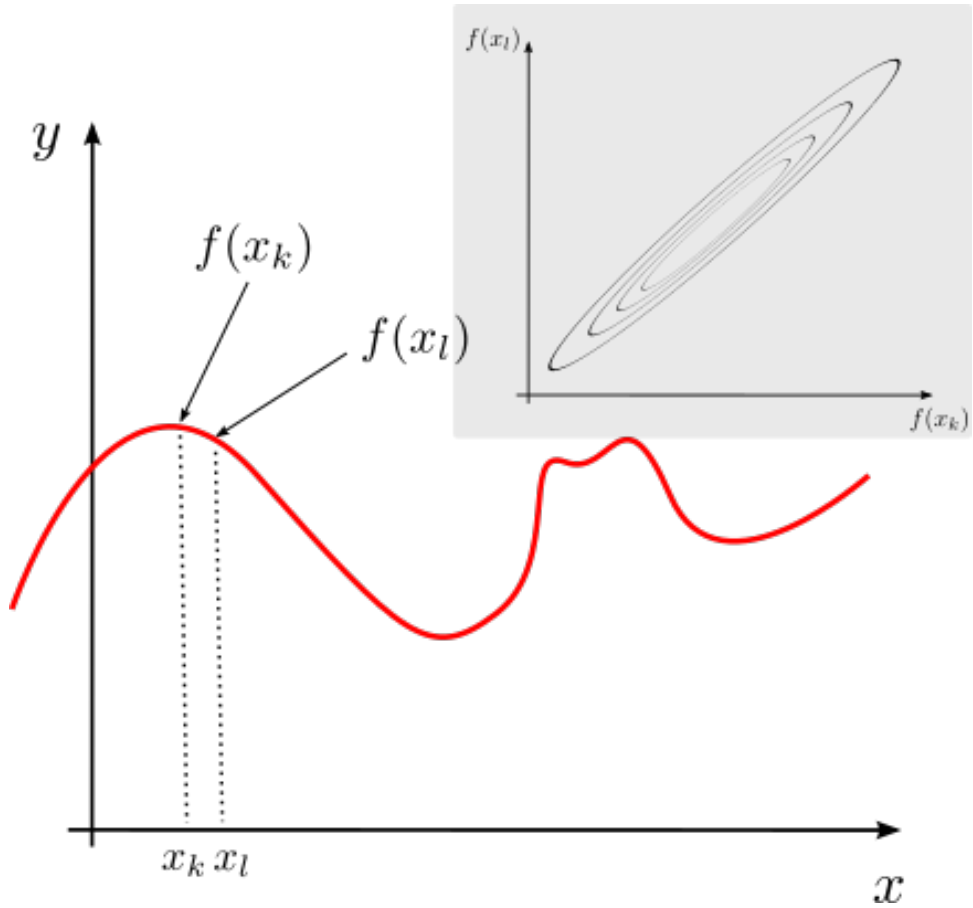
$$Cov\left(f(x_p), f(x_q)\right) = k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(x_p - x_q)^2\right)$$

- Nearby x's have highly correlated function ordinates $f(x)$.
- Extension to multiple covariates: $(x_p - x_q)$ replaced by $\|\mathbf{x}_p - \mathbf{x}_q\|$.

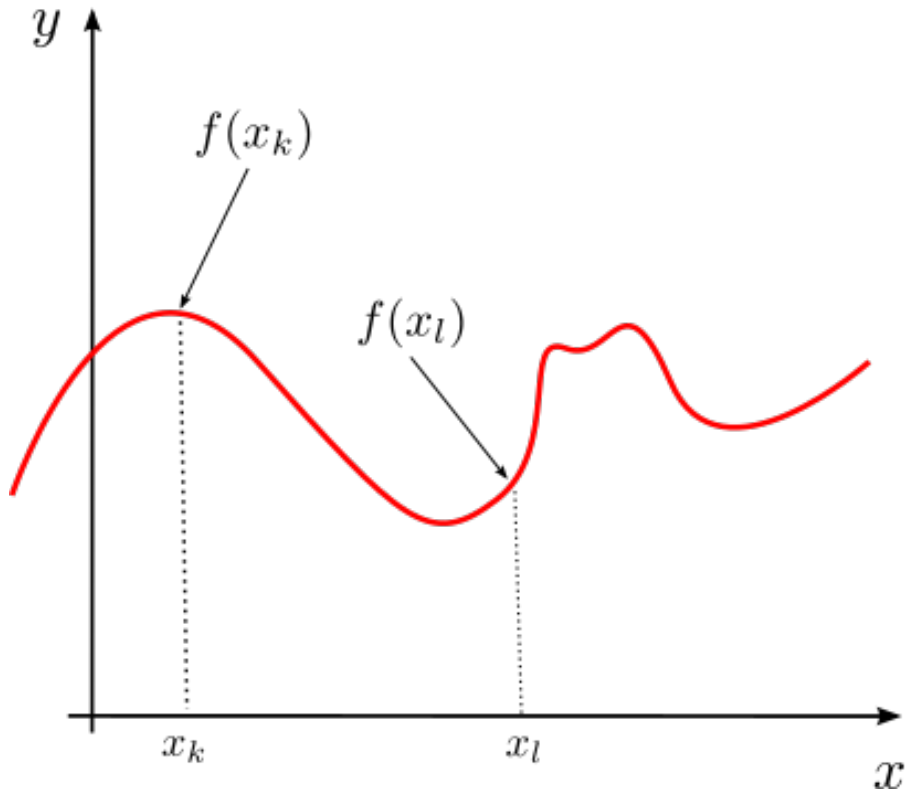
SMOOTH FUNCTION - POINTS NEARBY



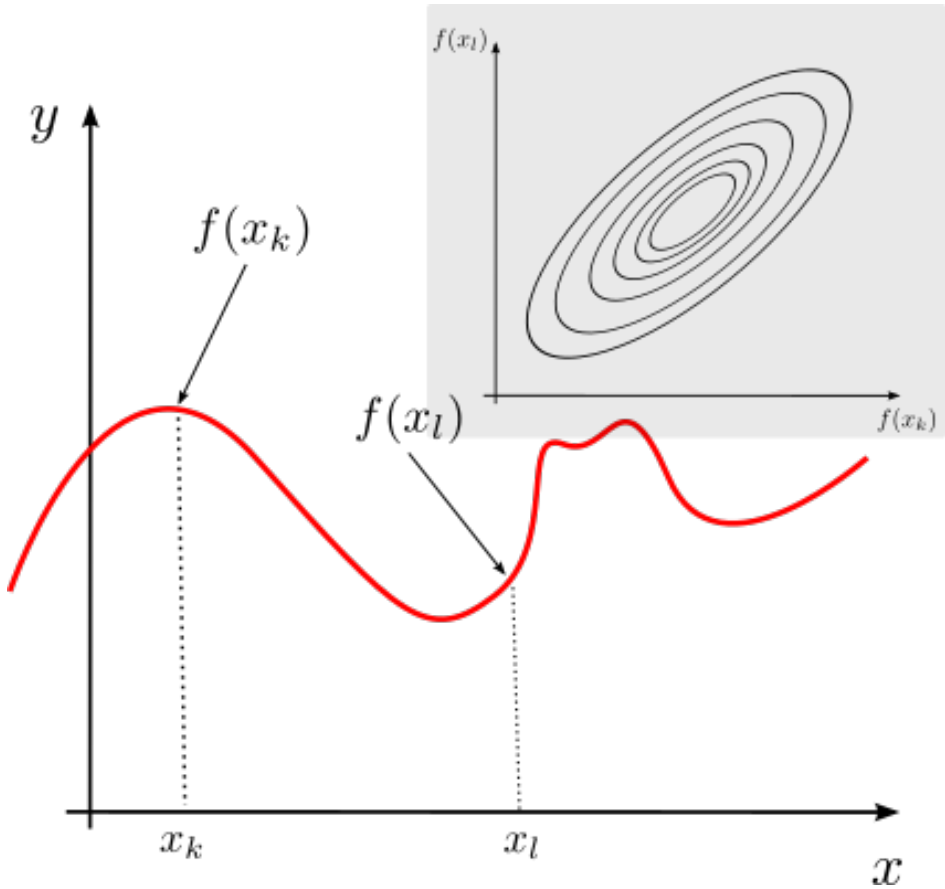
SMOOTH FUNCTION - POINTS NEARBY



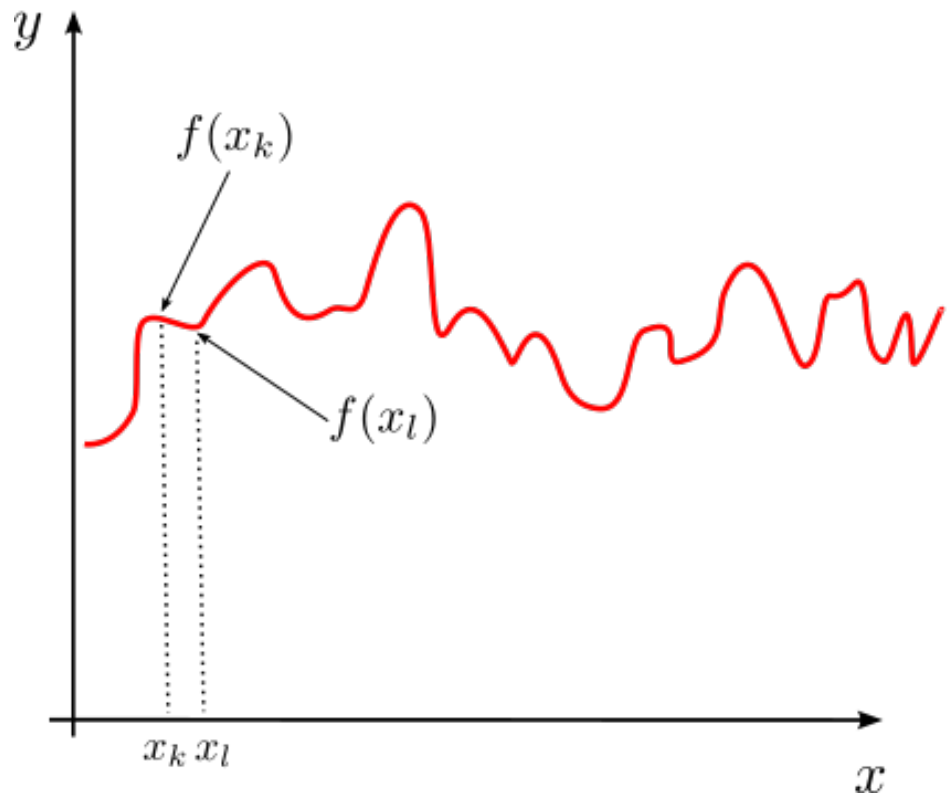
SMOOTH FUNCTION - POINTS FAR APART



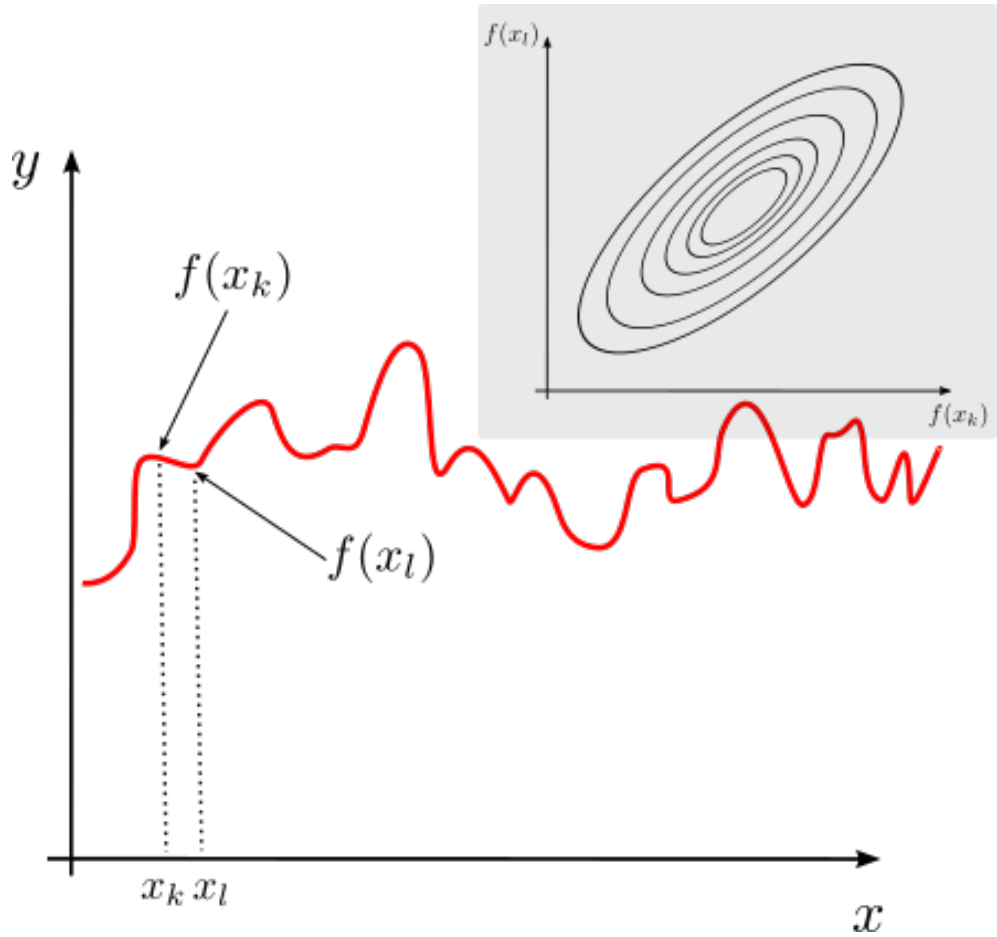
SMOOTH FUNCTION - POINTS FAR APART



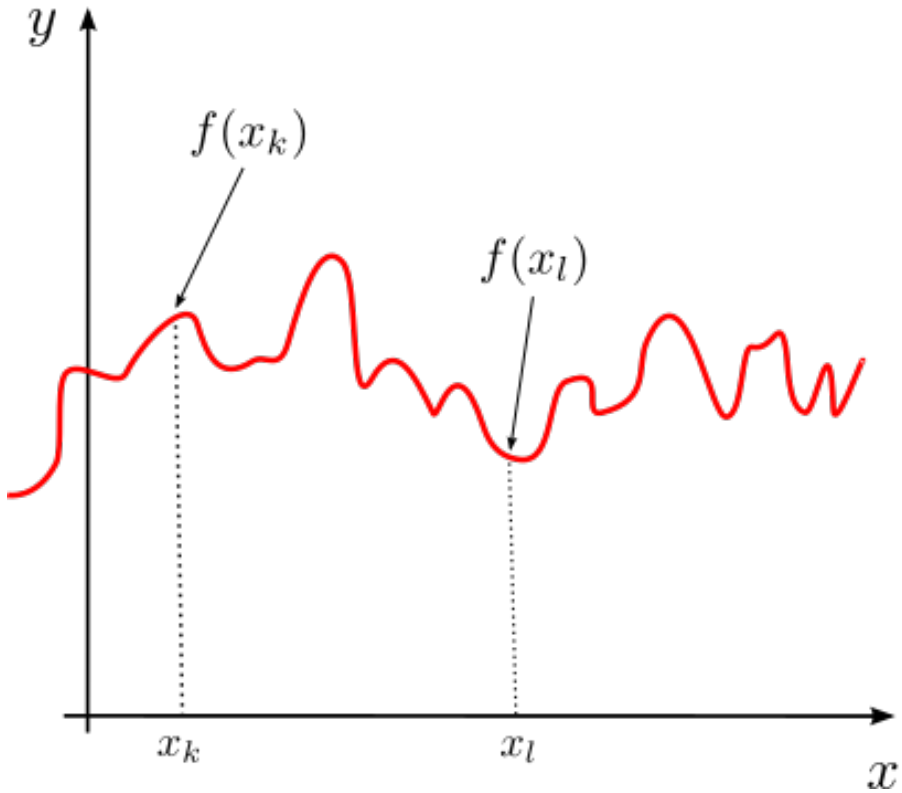
JAGGED FUNCTION - POINTS NEARBY



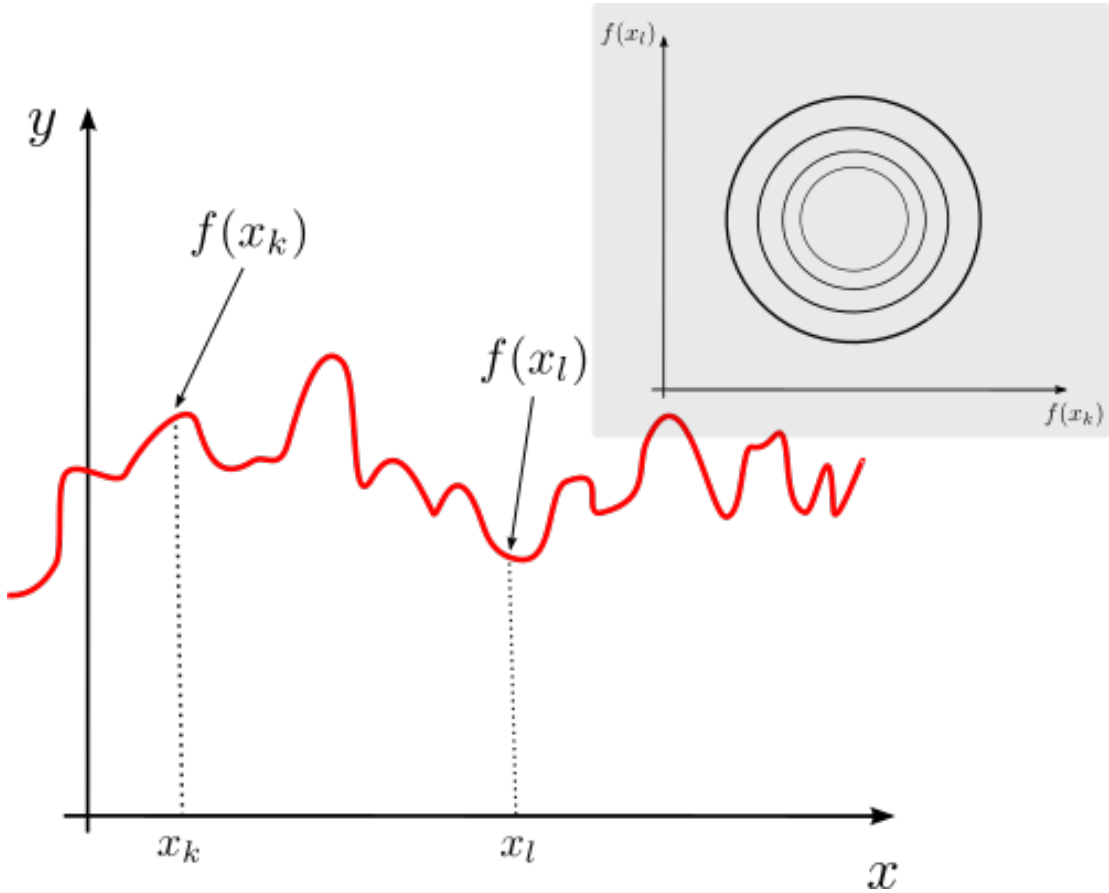
JAGGED FUNCTION - POINTS NEARBY



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). No need for a grid!
- 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:

$$m(x) = \sin(x)$$

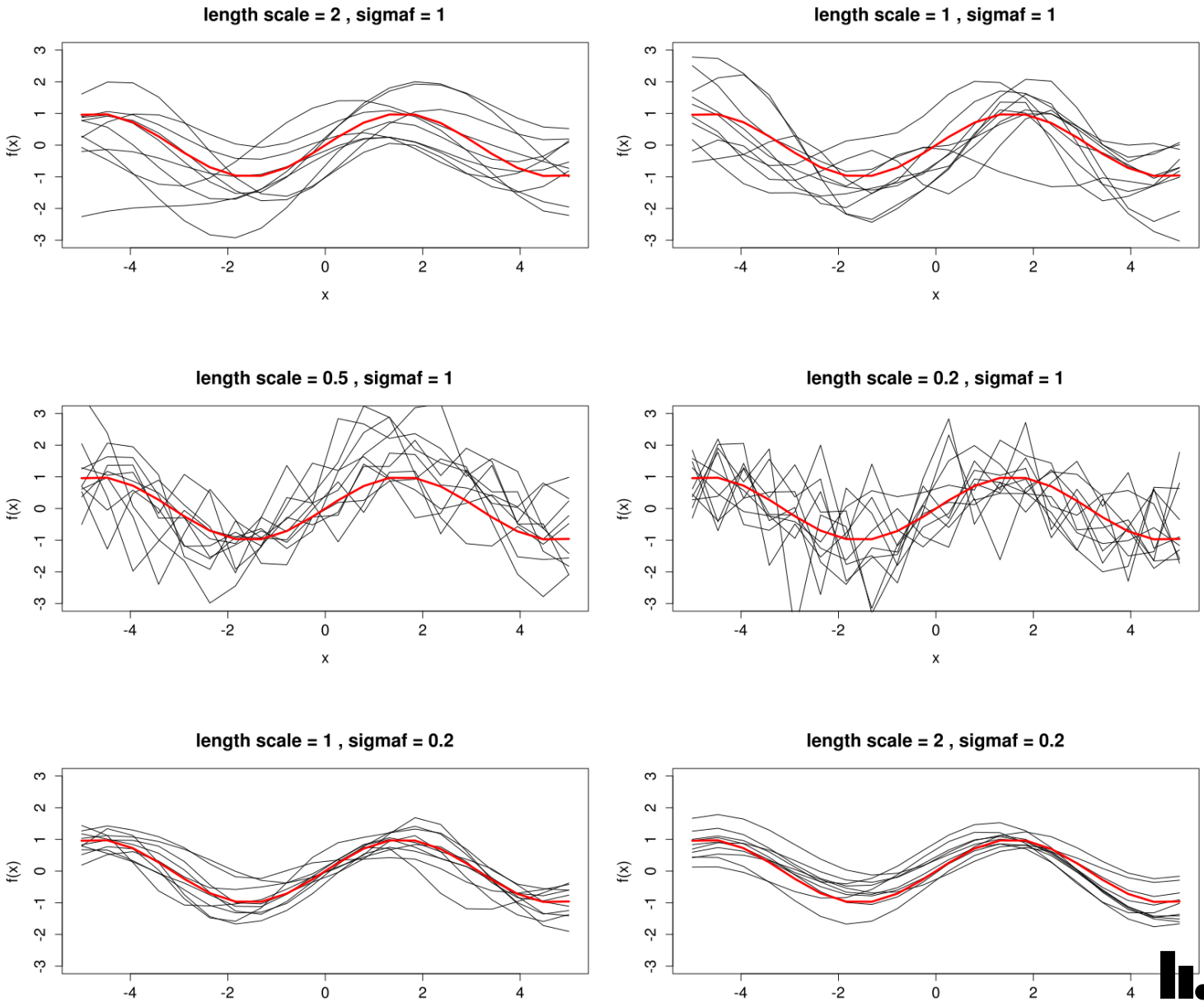
$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{x - x'}{\ell}\right)^2\right)$$

  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

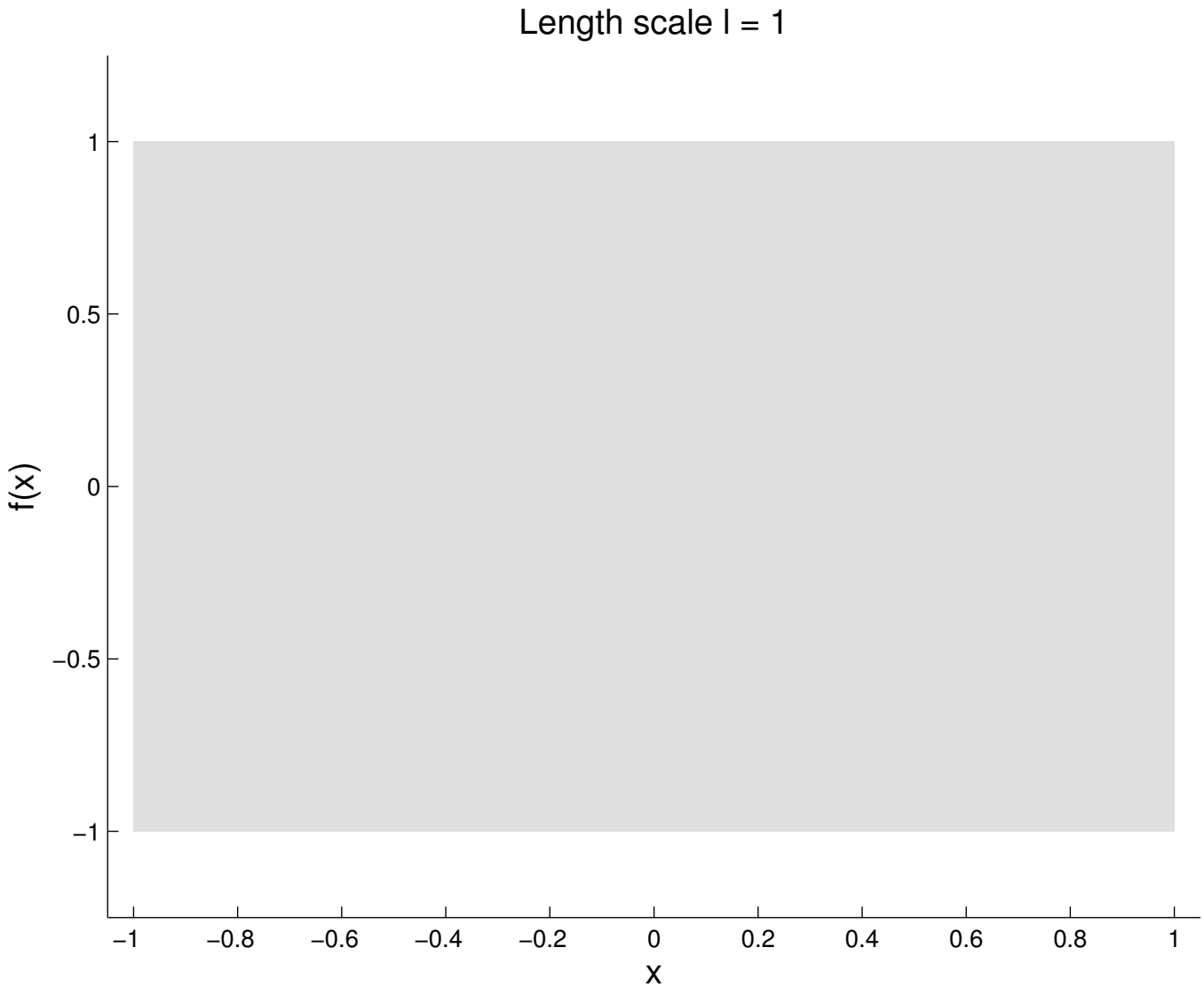
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

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

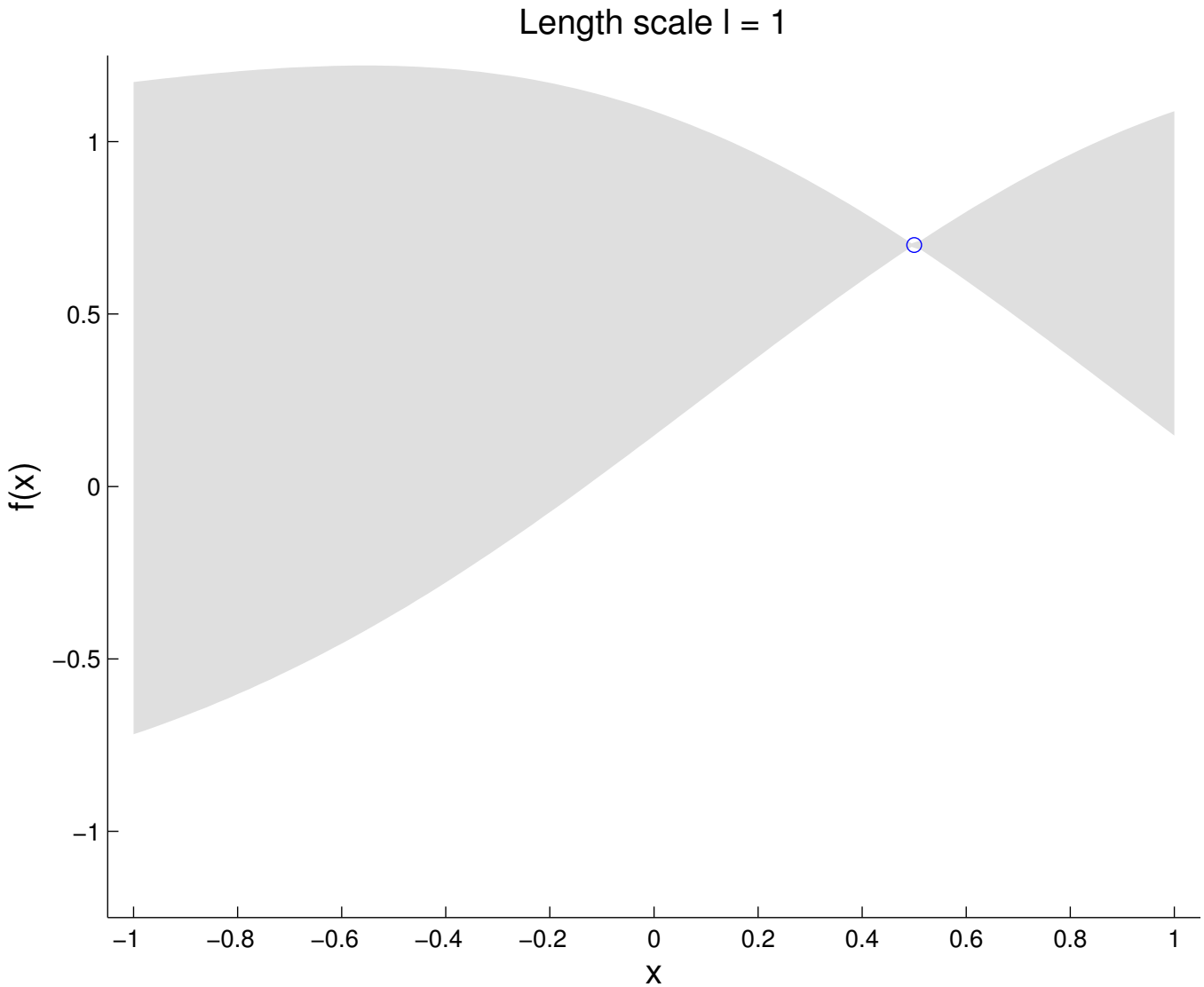
- Try my RStudio Manipulate code in the file `GaussianProcesses.R`.
- More intuition from the **conditional decomposition**

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

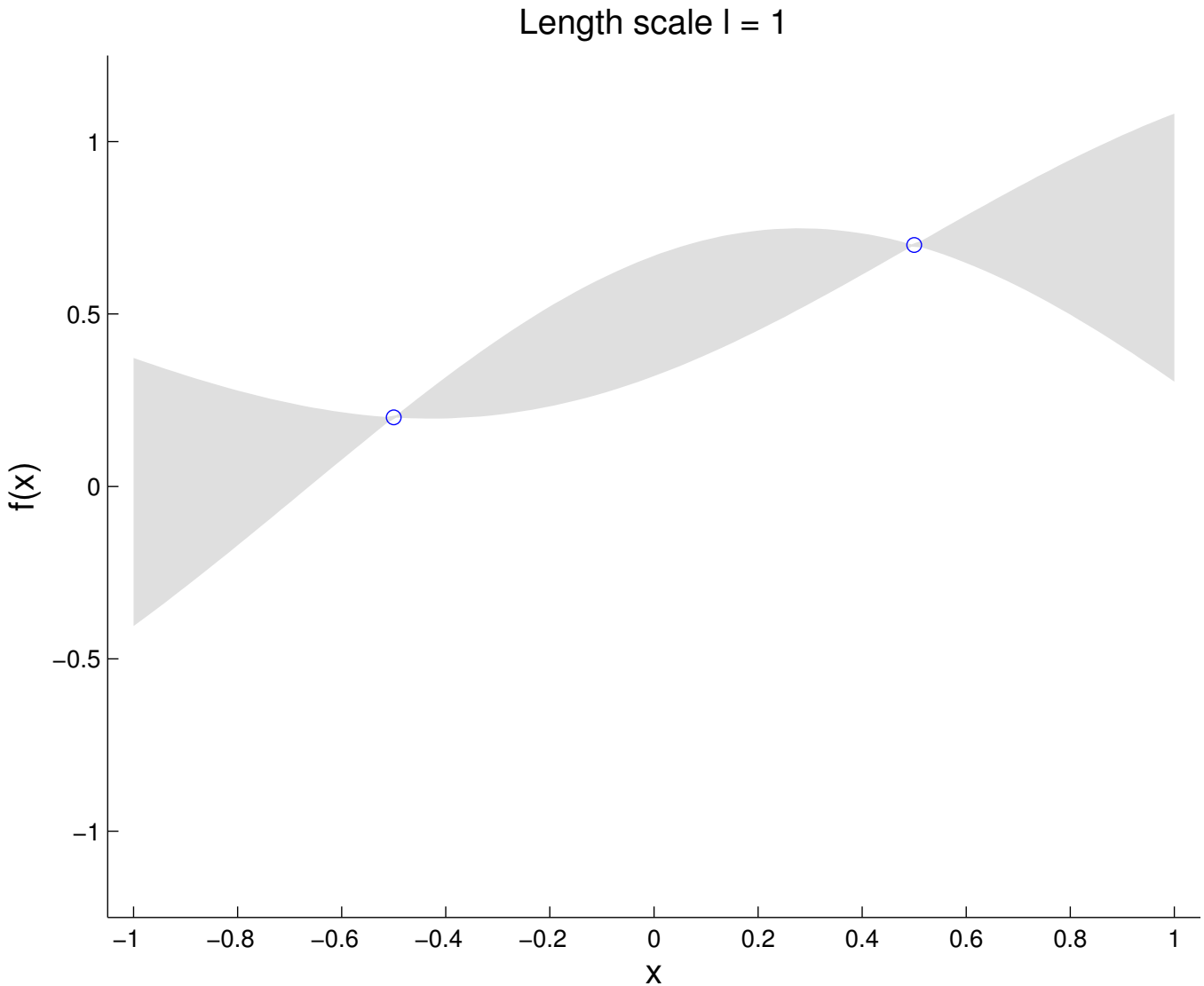
Density before first draw



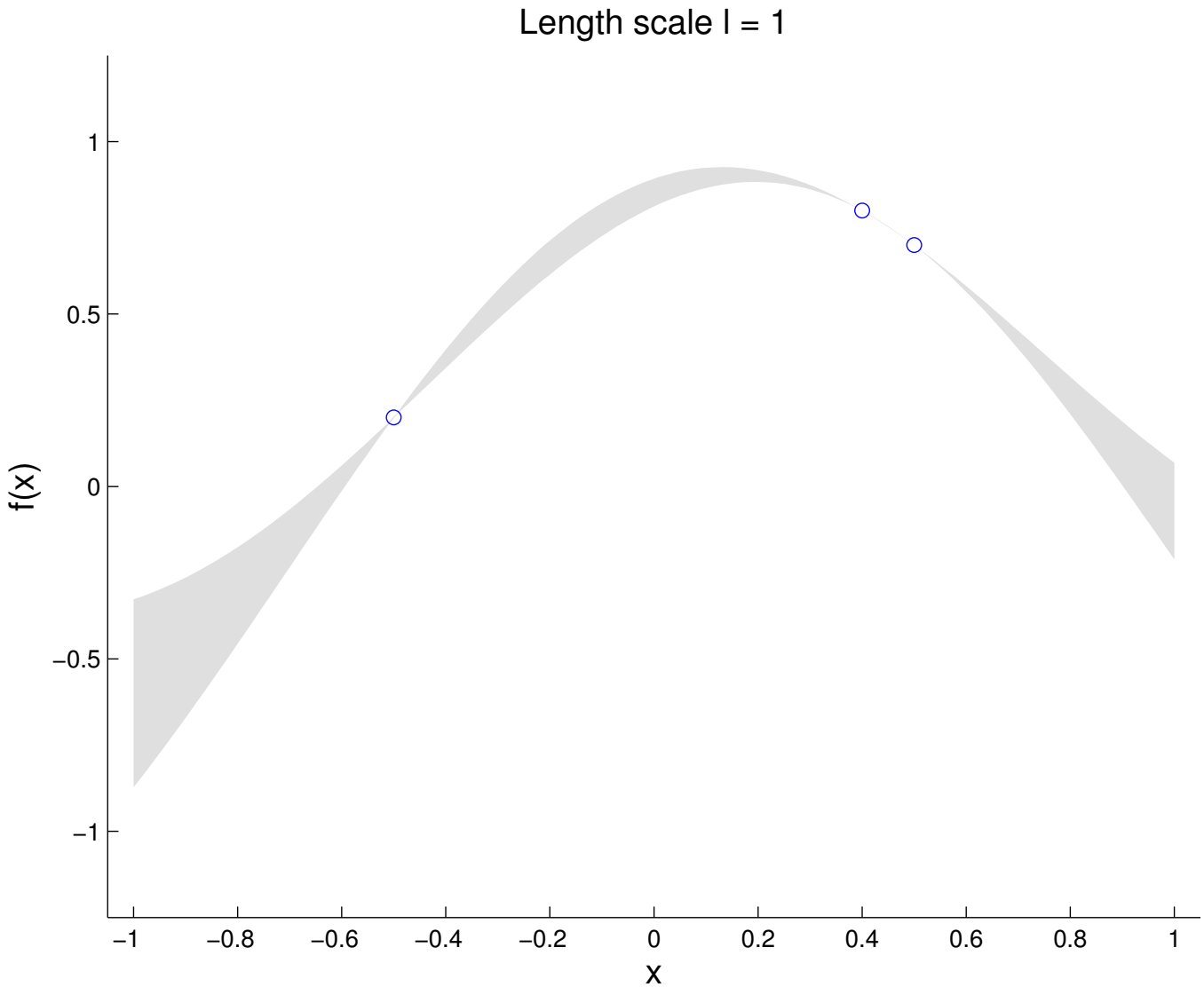
Density before second draw



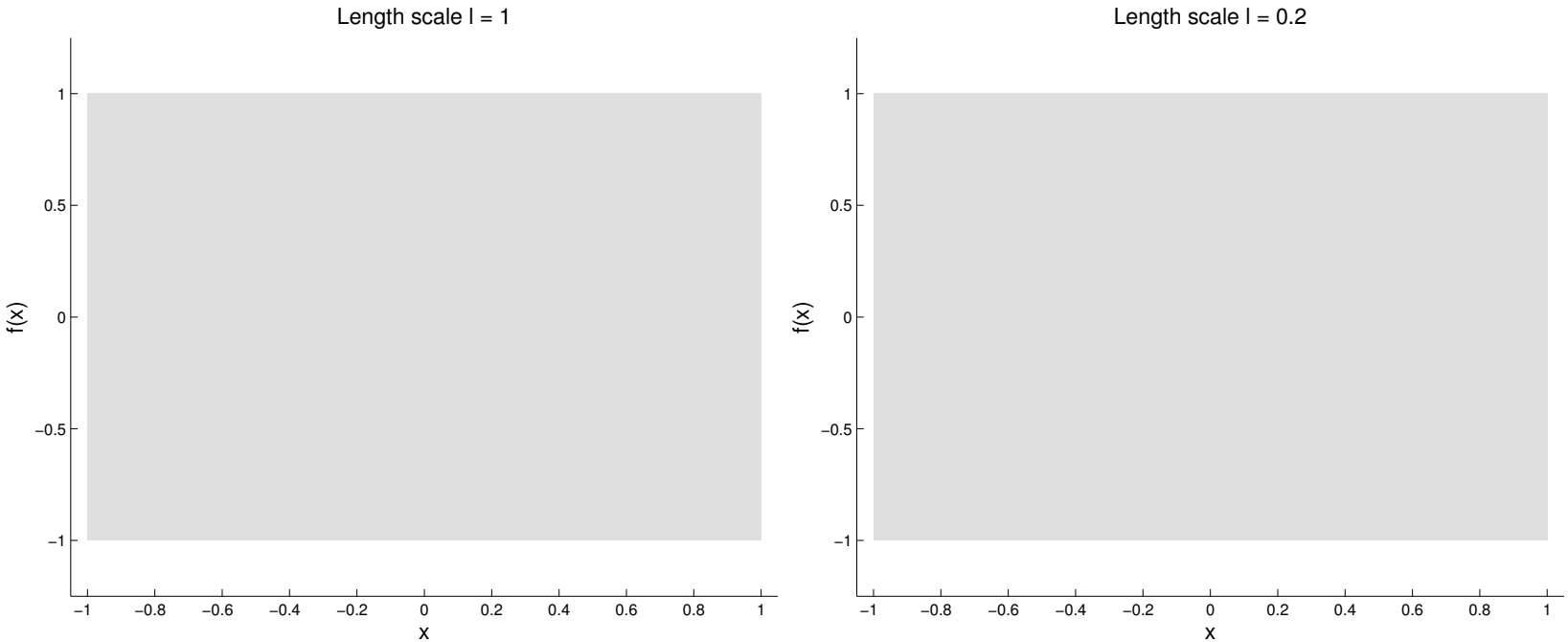
Density before third draw



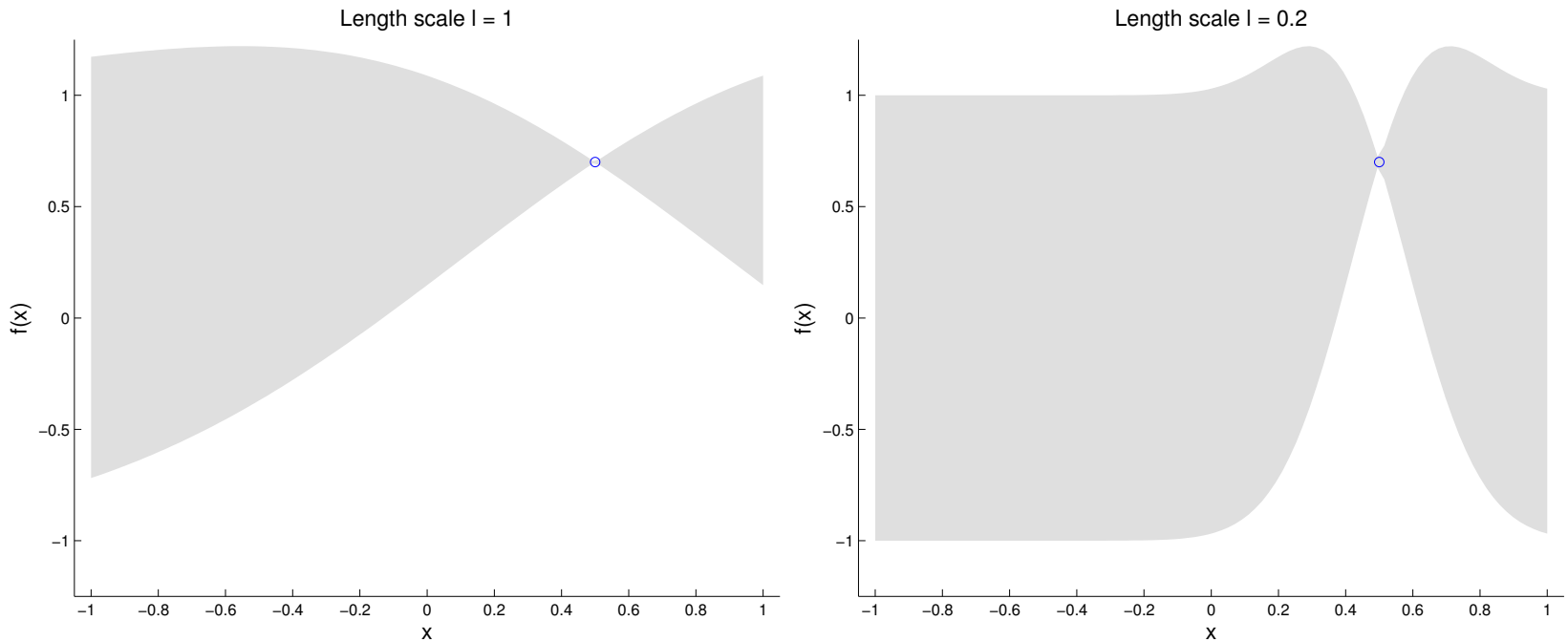
Density before fourth draw



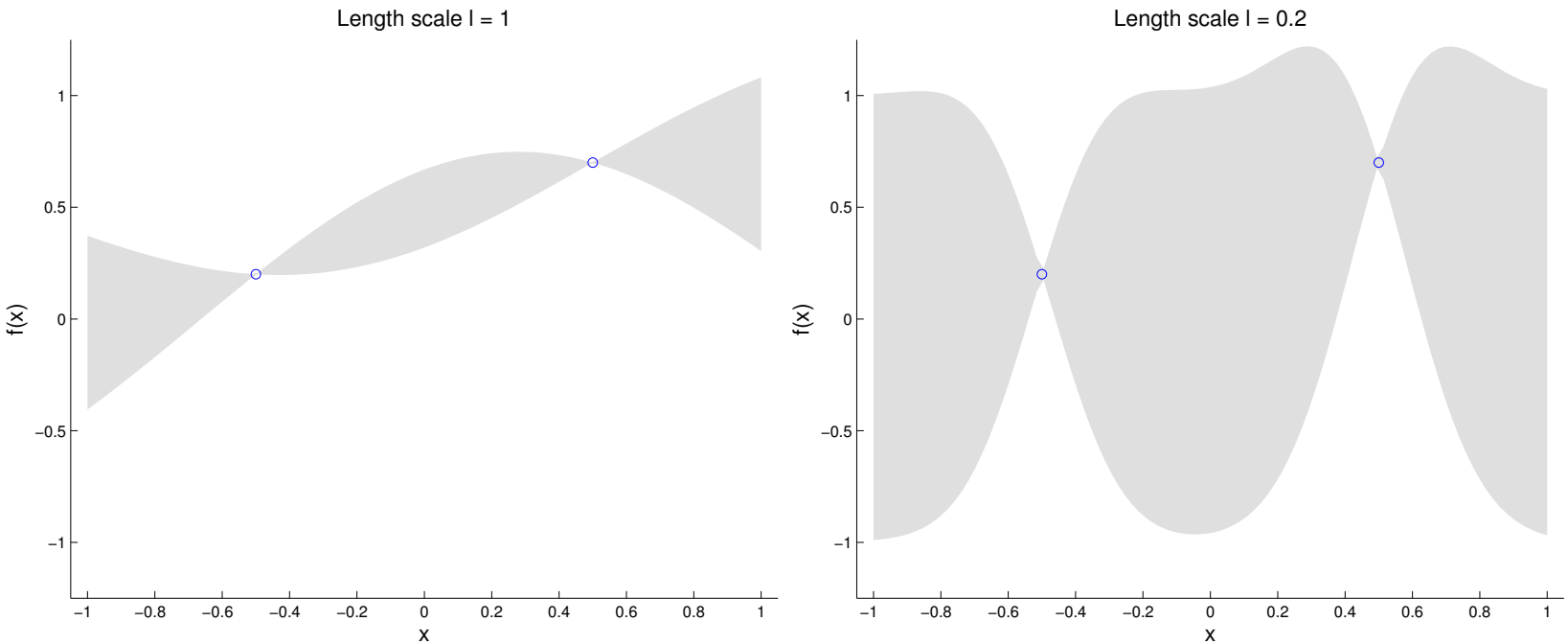
SIMULATION FROM L=1 VS L=0.2. BEFORE FIRST DRAW.



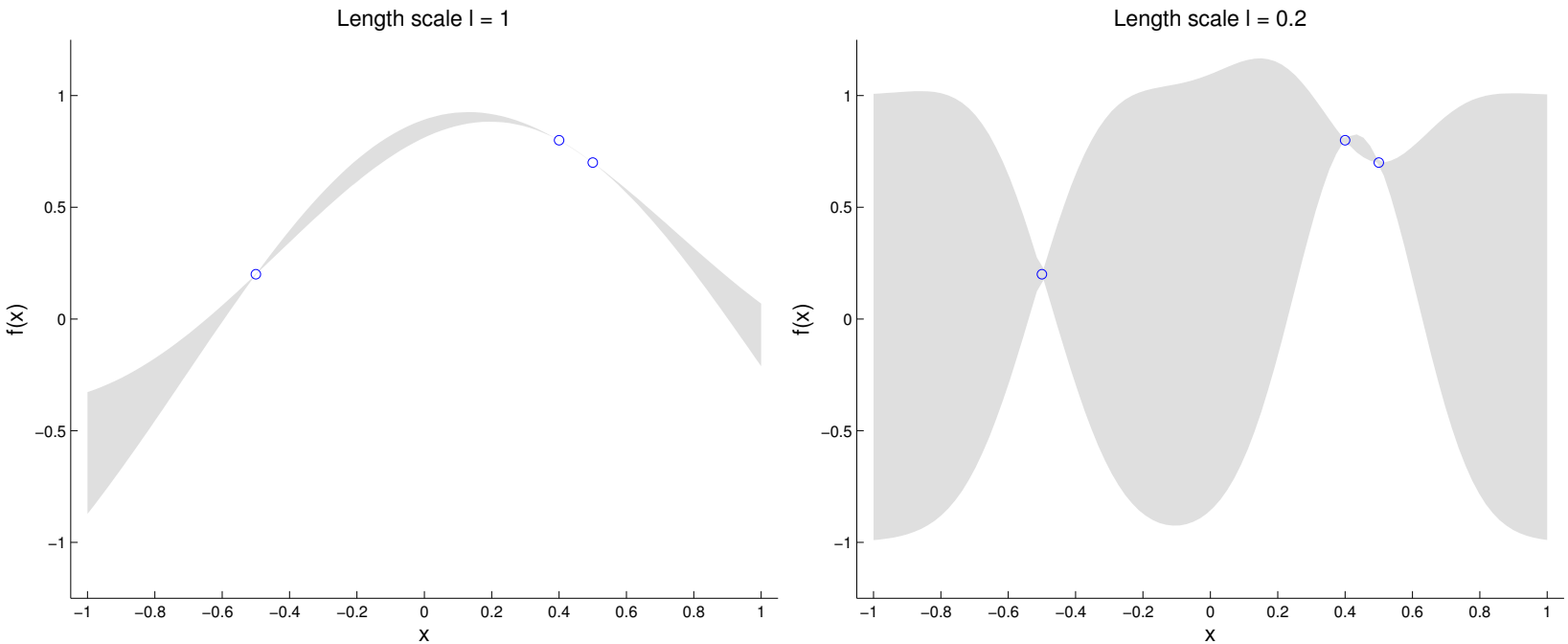
Simulation from l=1 vs l=0.2. Before second draw.



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



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



Posterior of a GP regression

- **Model**

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

- **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: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.

Posterior of a GP regression

- **Model**

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

- **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: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.
- Intermediate step: joint distribution of $\mathbf{y}$ and $\mathbf{f}_*$

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

Posterior of a GP regression

- **Model**

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

- **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: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.
- Intermediate step: joint distribution of $\mathbf{y}$ and $\mathbf{f}_*$

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

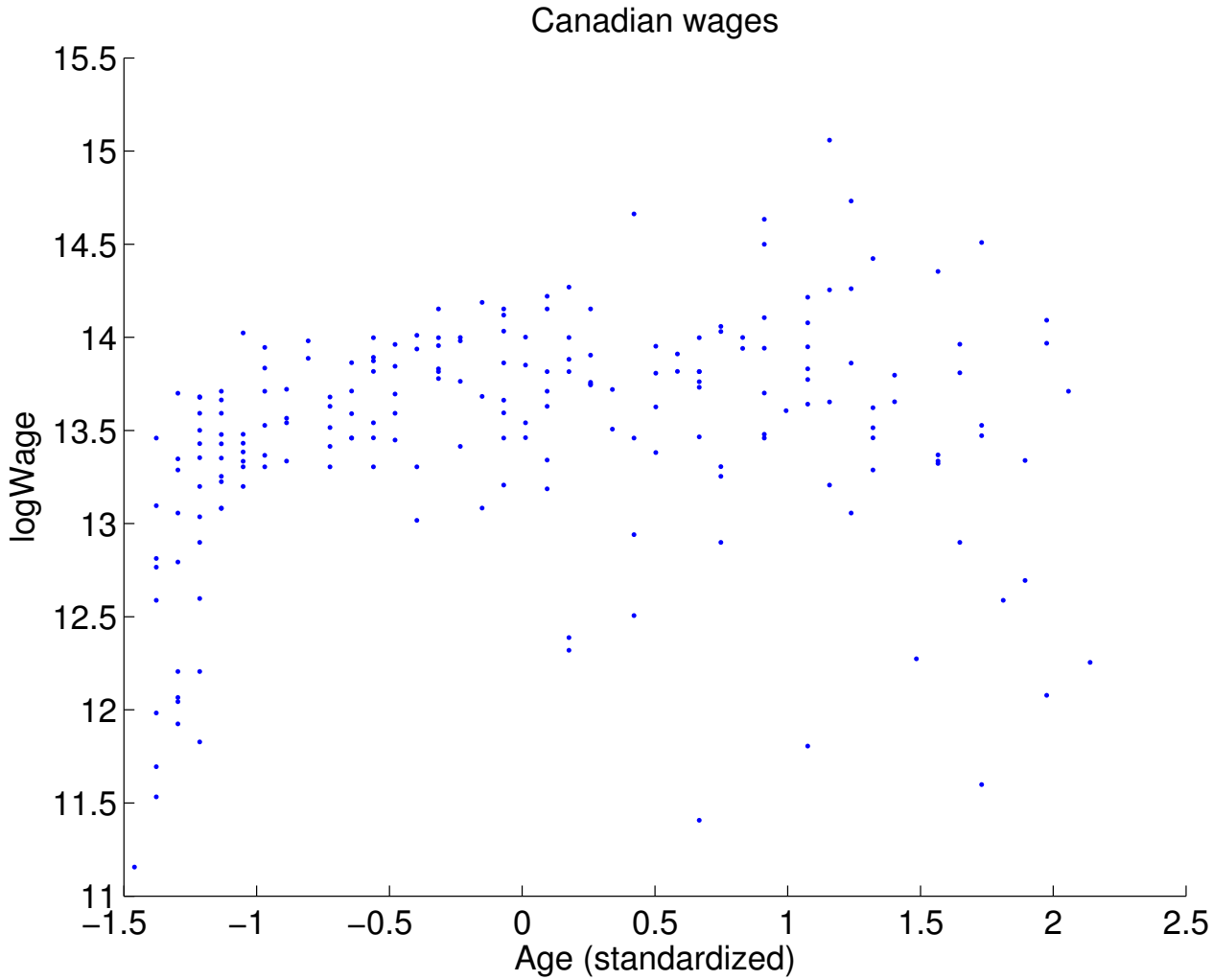
- **Posterior**

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

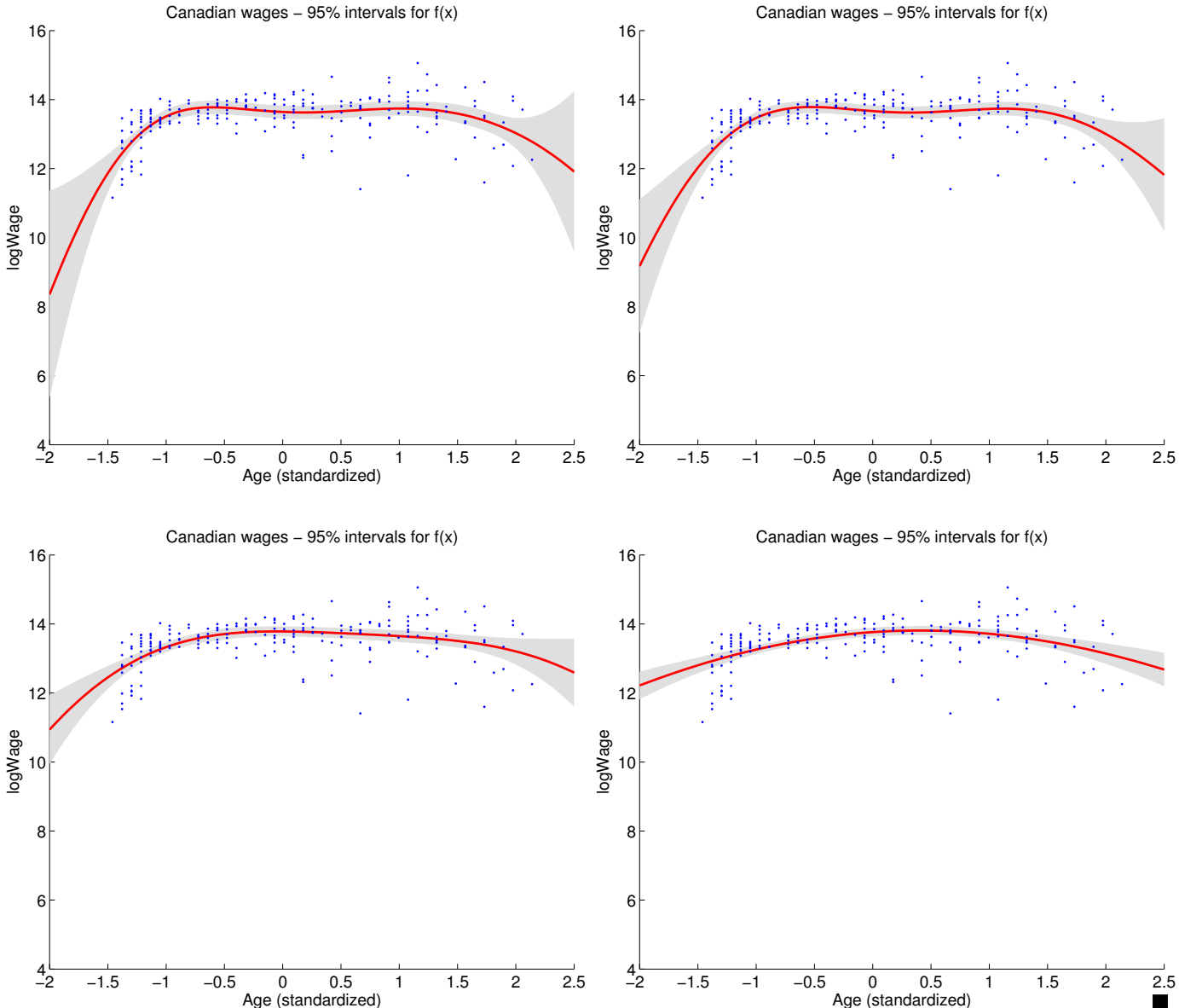
$$\bar{\mathbf{f}}_* = K(\mathbf{x}_*, \mathbf{x}) \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} \mathbf{y}$$

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

Example - Canadian wages



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



Commonly used covariance kernels

- Let $r = \|x - x'\|$. All kernels can be scaled by $\sigma_f > 0$.
- **Squared exponential (SE)** ($\ell > 0$)

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

  - Infinitely mean square differentiable. Very smooth.

- **Matérn** ($\ell > 0$, $\nu > 0$)

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

  - $\nu = 3/2$ (first-order continuous) and $\nu = 5/2$ (second-order continuous) are common choices.
  - As $\nu \to \infty$, Matérn's kernel approaches SE kernel.

More on kernels

- 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.
- **Factor kernels**: $M = \Lambda\Lambda^T + \Psi$, where Λ is $D \times k$ for low rank k.
- Length-scales $\ell(\mathbf{x})$ that vary with $\mathbf{x}$. Gibbs kernel in RW Eq. 4.32. **Adaptive smoothness**.
- Kernels are often combined into **composite kernels**. Sum of kernels is a kernel. Product of kernels is a kernel.

Bayesian inference for hyperparameters

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

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

- If the hyperparameters are unknown, just compute the posterior

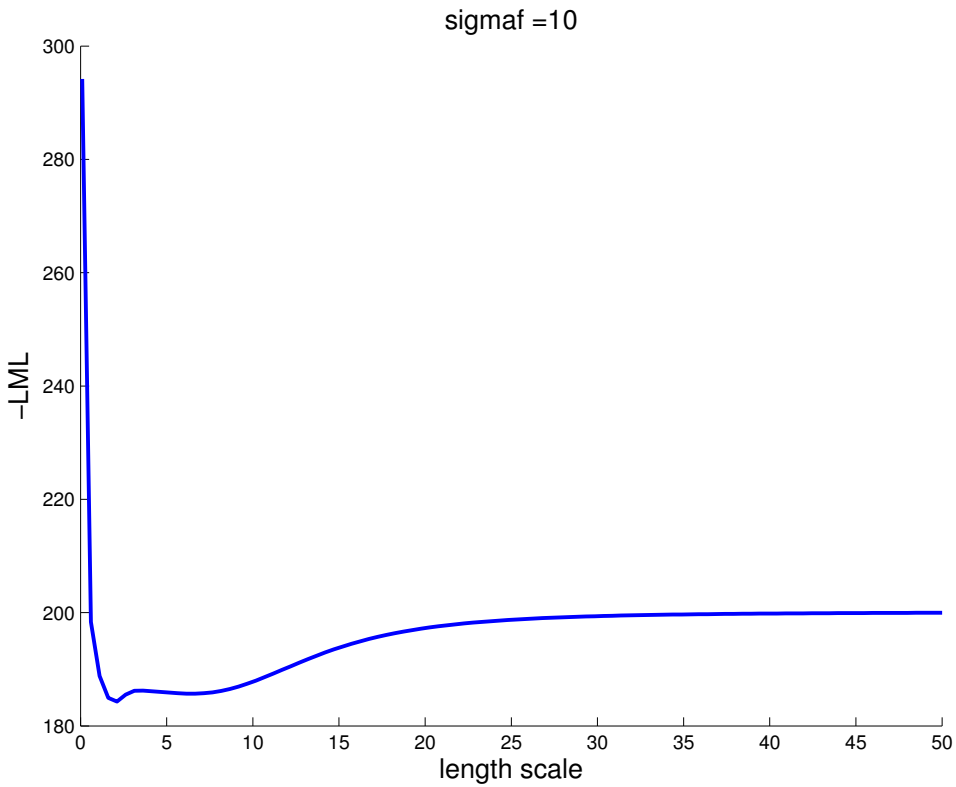
$$p(\theta|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \theta)p(\theta|\mathbf{X}).$$

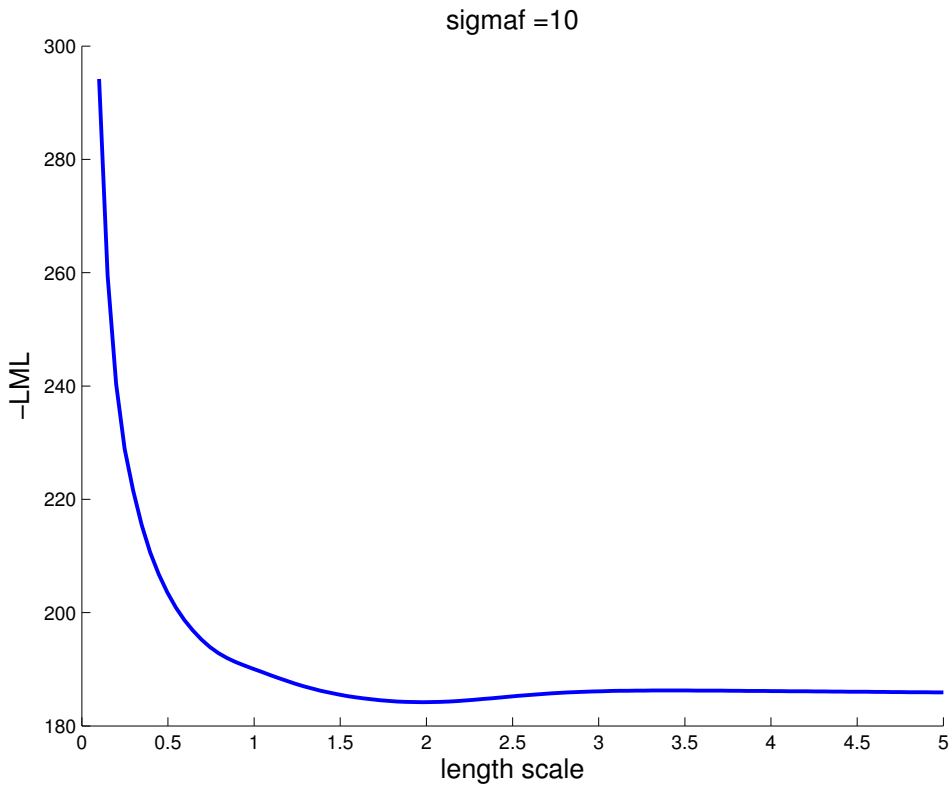
- For Gaussian process regression [since $\mathbf{y}|\mathbf{X}, \theta \sim N(0, K + \sigma_n^2 I)$]

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

- **Empirical Bayes**: estimate θ by minimizing $-\log p(\mathbf{y}|\mathbf{X}, \theta)$.

CANADIAN WAGES - LML DETERMINATION OF ℓ





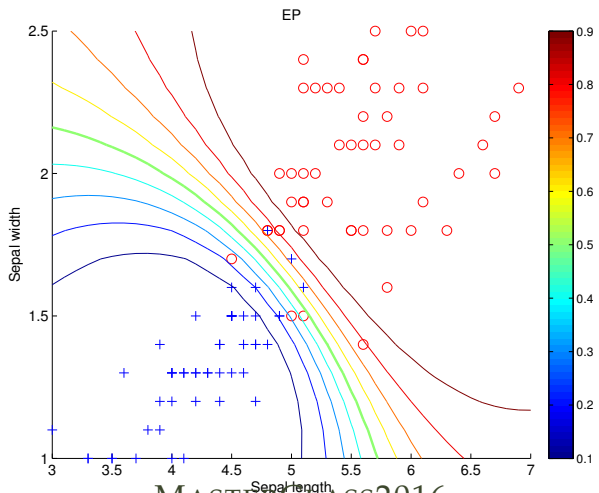
Classification

- **Classification**: **binary response** $y \in \{-1, 1\}$ (or multi-class $y \in \{1, 2, ..., C\}$) explained/predicted by covariates/features $\mathbf{x}$.
- Example: linear logistic regression

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

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

- $\lambda(z)$ 'squashes' the linear prediction $\mathbf{x}^T\beta \in \mathbb{R}$ into $\lambda(\mathbf{x}^T\beta) \in [0, 1]$.
- **Decision boundaries**: level contours of $p(y|\mathbf{x})$ over $\mathbf{x}$-space.



GP CLASSIFICATION

- **Linear logistic regression**

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

has linear decision boundaries (conditional on β).

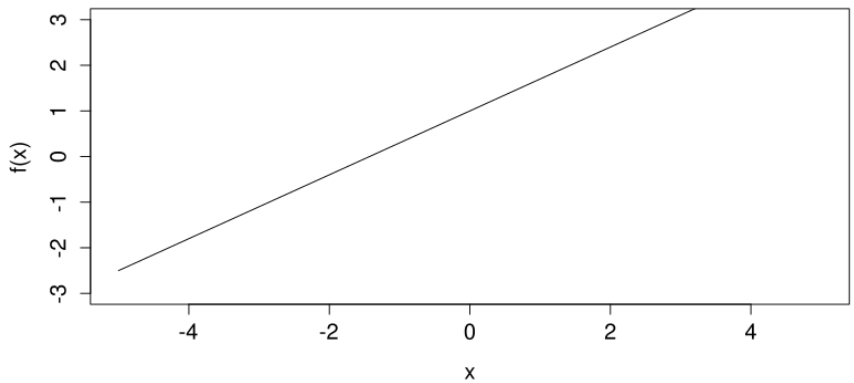
- Obvious **GP extension**: replace $\mathbf{x}^T\beta$ by

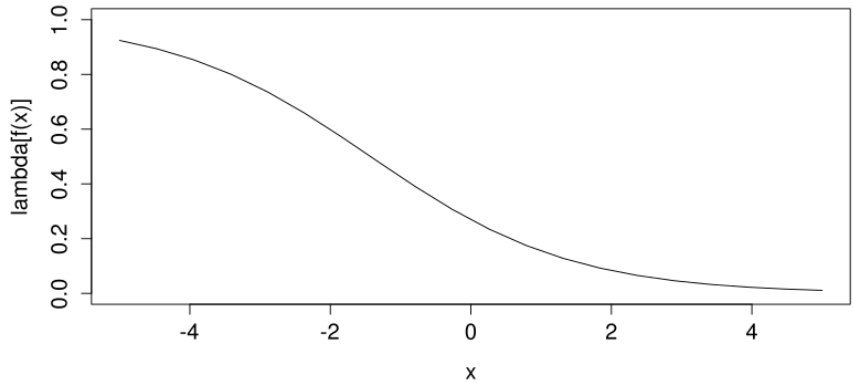
$$f(\mathbf{x}) \sim GP(0, k(\mathbf{x}, \mathbf{x}'))$$

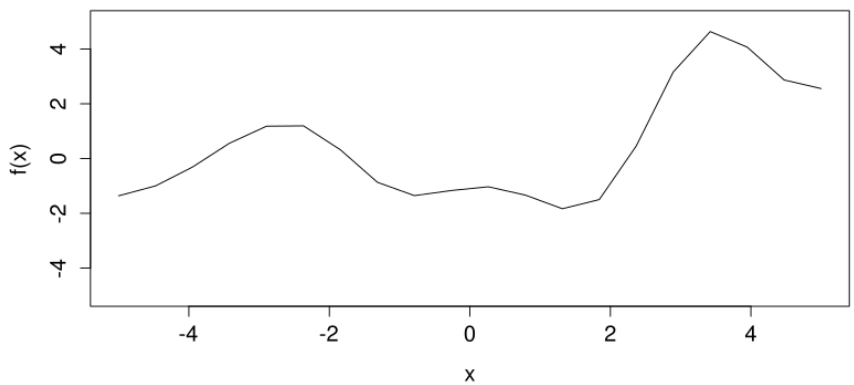
and squash f through logistic function

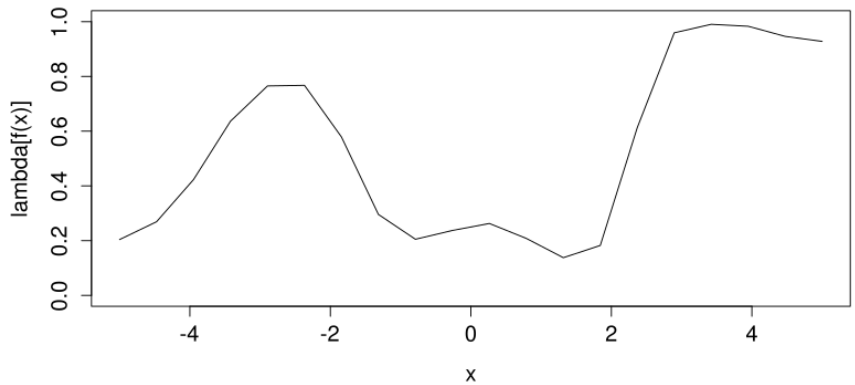
$$Pr(y = 1|\mathbf{x}) = \lambda(f(\mathbf{x}))$$

SQUASHING F


f(x)
-3
-2
-1
0
1
2
3
-4
-2
0
2
4
x


lambda[f(x)]
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
x


f(x)
-4
-2
0
2
4
-4
-2
0
2
4
x


lambda[f(x)]
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
x

The Laplace approximation

- Approximates $p(\mathbf{f}|\mathbf{X},\mathbf{y})$ with $N(\hat{\mathbf{f}},\mathbf{A}^{-1})$, where $\hat{\mathbf{f}}$ is the posterior mode and $\mathbf{A}$ is the negative Hessian of the log posterior at $\mathbf{f}=\hat{\mathbf{f}}$.
- The log posterior is (proportional to)

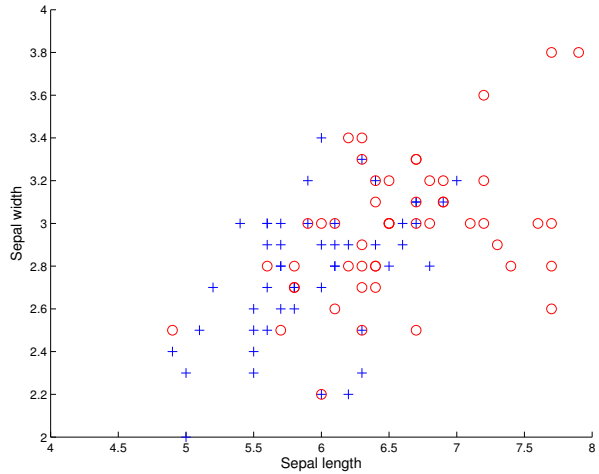
$$\begin{aligned}\Psi(\mathbf{f}) &= \log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{f}|\mathbf{X}) \\ &= \log p(\mathbf{y}|\mathbf{f}) - \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi\end{aligned}$$

- Differentiating wrt $\mathbf{f}$

$$\begin{aligned}\nabla\Psi(\mathbf{f}) &= \nabla \log p(\mathbf{y}|\mathbf{f}) - \mathbf{K}^{-1}\mathbf{f} \\ \nabla\nabla\Psi(\mathbf{f}) &= \nabla\nabla \log p(\mathbf{y}|\mathbf{f}) - \mathbf{K}^{-1} = -\mathbf{W} - \mathbf{K}^{-1}\end{aligned}$$

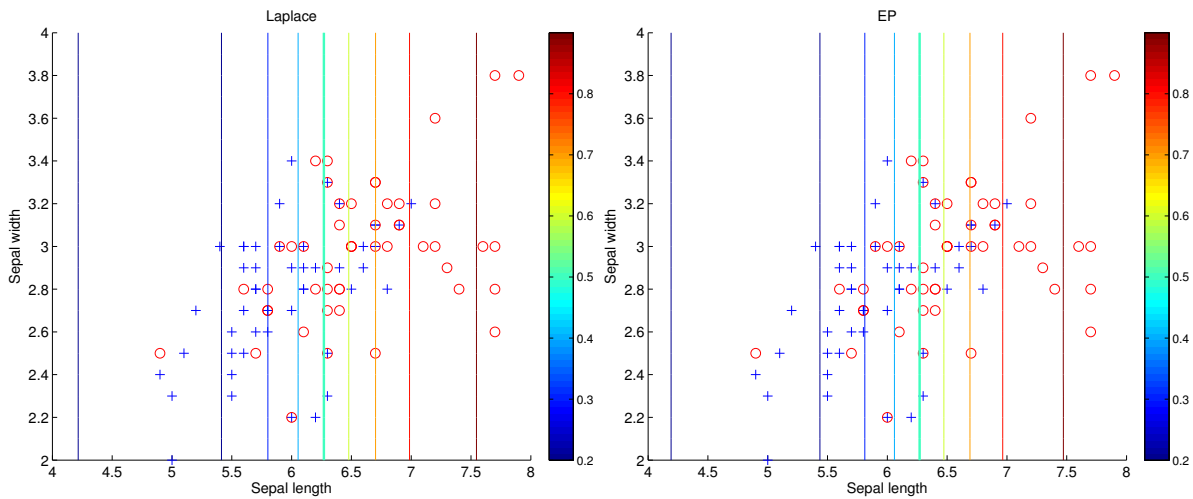
  where $\mathbf{W}$ is a diagonal matrix since each y_i only depends on its f_i.
- Use **Newton's method** to iterate to the mode.

IRIS DATA - SEPAL - SE KERNEL WITH ARD

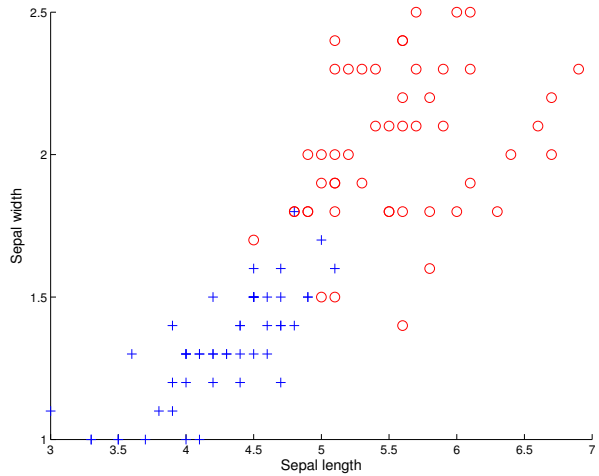


Laplace: $\hat{\ell}_1 = 1.7214, \hat{\ell}_2 = 185.5040, \sigma_f = 1.4361$

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

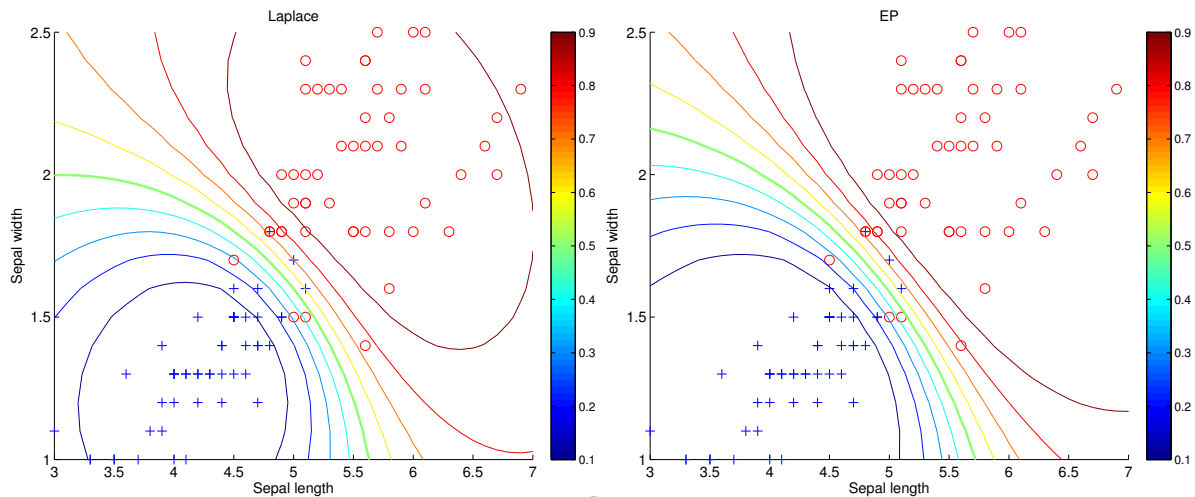


IRIS DATA - PETAL - SE KERNEL WITH ARD

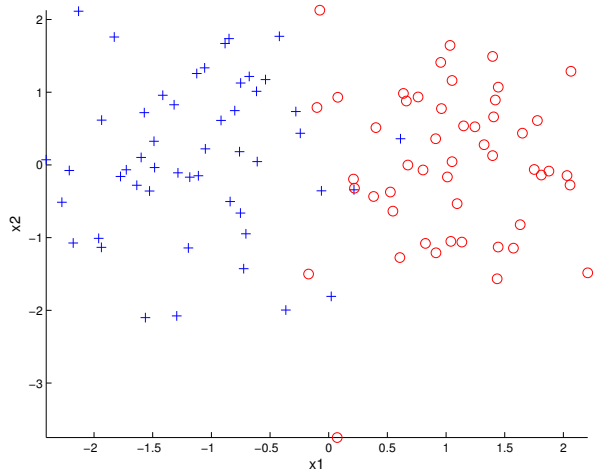


Laplace: $\hat{\ell}_1 = 1.7606, \hat{\ell}_2 = 0.8804, \sigma_f = 4.9129$

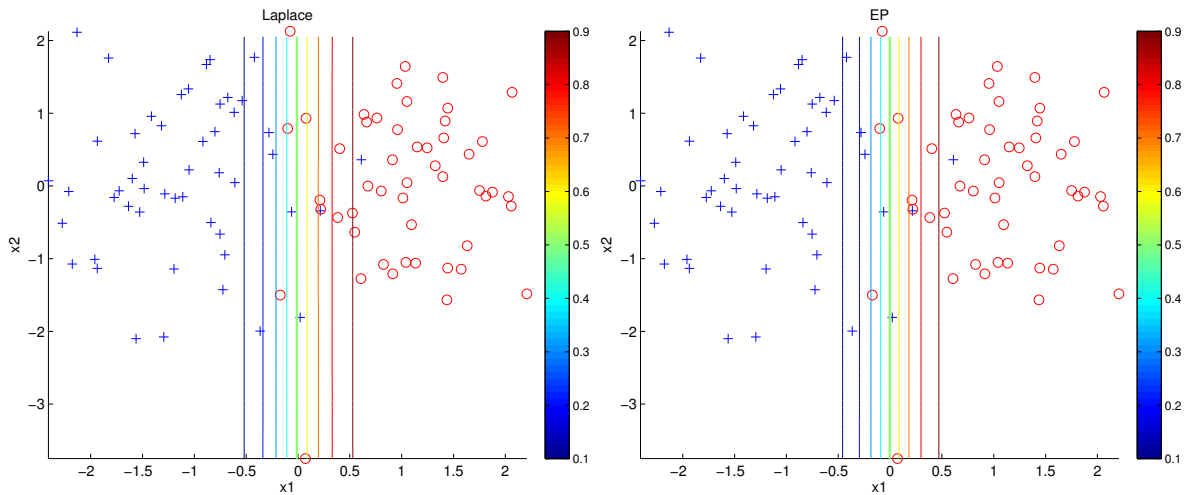
EP: $\hat{\ell}_1 = 2.1139, \hat{\ell}_2 = 1.0720, \sigma_f = 5.3369$



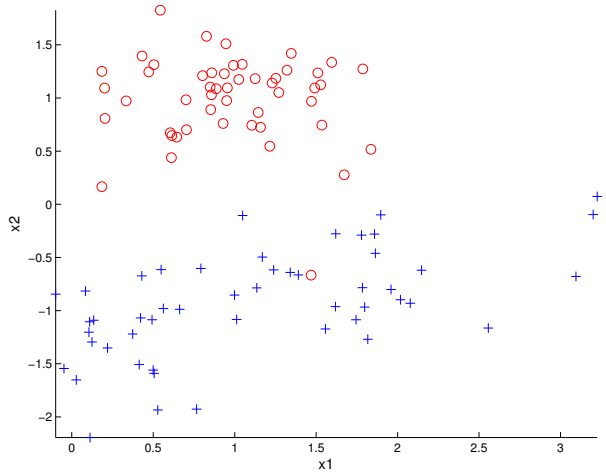
TOY DATA 1 - SE KERNEL WITH ARD



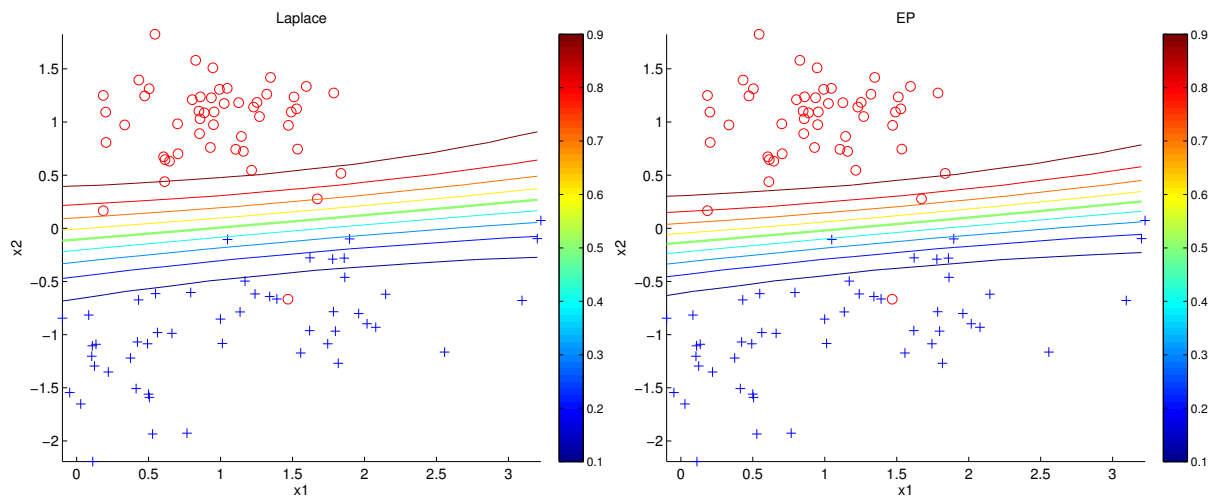
EP: $\hat{\ell}_1 = 2.4503, \hat{\ell}_2 = 721.7405, \sigma_f = 4.7540$



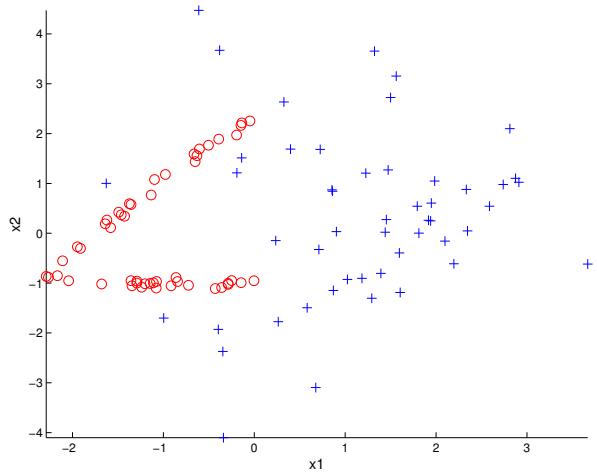
TOY DATA 2 - SE KERNEL WITH ARD



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

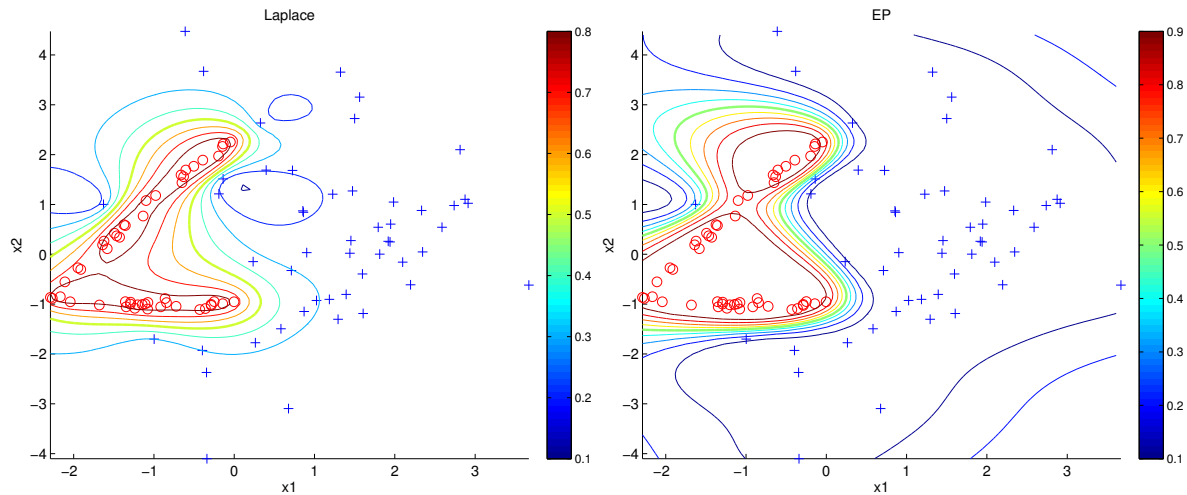


TOY DATA 3 - SE KERNEL WITH ARD



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

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



GPs are ubiquitous

- **VARs** with **nonparametric steady state**

$$\begin{pmatrix} y_{1t} \\ \vdots \\ y_{pt} \end{pmatrix} = \begin{pmatrix} \mu_1(t) \\ \vdots \\ \mu_p(t) \end{pmatrix} + \Pi \left(\begin{pmatrix} y_{1t} \\ \vdots \\ y_{pt} \end{pmatrix} - \begin{pmatrix} \mu_1(t) \\ \vdots \\ \mu_p(t) \end{pmatrix} \right) + \begin{pmatrix} \varepsilon_{1t} \\ \vdots \\ \varepsilon_{pt} \end{pmatrix}$$

  where $\mu_j(t)$ is the steady state of $\{y_{jt}\}_{t=1}^{T}$, and $\mu_j(t) \sim GP$ a priori.

- **Nonparametric system identification** in state-space models

$$\begin{aligned} y_t &= h(x_t) + v_t \\ x_t &= \rho x_{t-1} + w_t \end{aligned}$$

  and $h \sim GP$.

- **Hemodynamics in functional MRI** (brain imaging).
  $h \sim GP(\mu_{Physio}, K)$, where μ_{Physio} is a simplified physiological model for blood flows.

Gaussian Process Optimization (GPO)

- **Aim**: minimization of **expensive** function

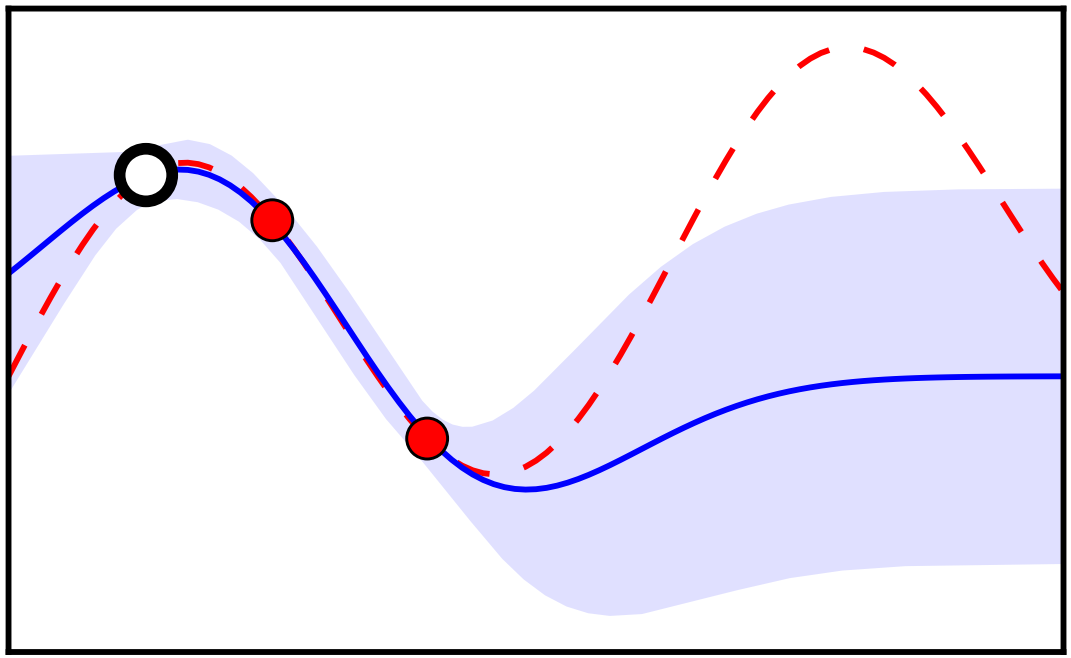
$$\text{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

- Typical applications: **hyperparameter estimation**.

- **GPO idea**:
  - Assign GP prior to the unknown function f.
  - Evaluate the function at some values $x_1, x_2, ..., x_n$.
  - Update to posterior $f|x_1, ..., x_n \sim GP(\mu, K)$. Noise-free model.
  - Use the GP posterior of f to find a new evaluation point x_{n+1}. **Explore** vs **Exploit**.
  - Iterate until the change in optimum is lower that some tolerance.

- **Bayesian Optimization**. Bayesian Numerics. Probabilistic numerics.

Explore-Exploit Illustration



Acquisition functions

- **Probability of Improvement** (PI)

$$a_{PI}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) \equiv \Pr\left(f(\mathbf{x}) < f(\mathbf{x}_{best})\right) = \Phi(\gamma(\mathbf{x}))$$

where

$$\gamma(\mathbf{x}) = \frac{f(\mathbf{x}_{best}) - \mu\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)}{\sigma\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)}$$

- **Expected Improvement** (EI)

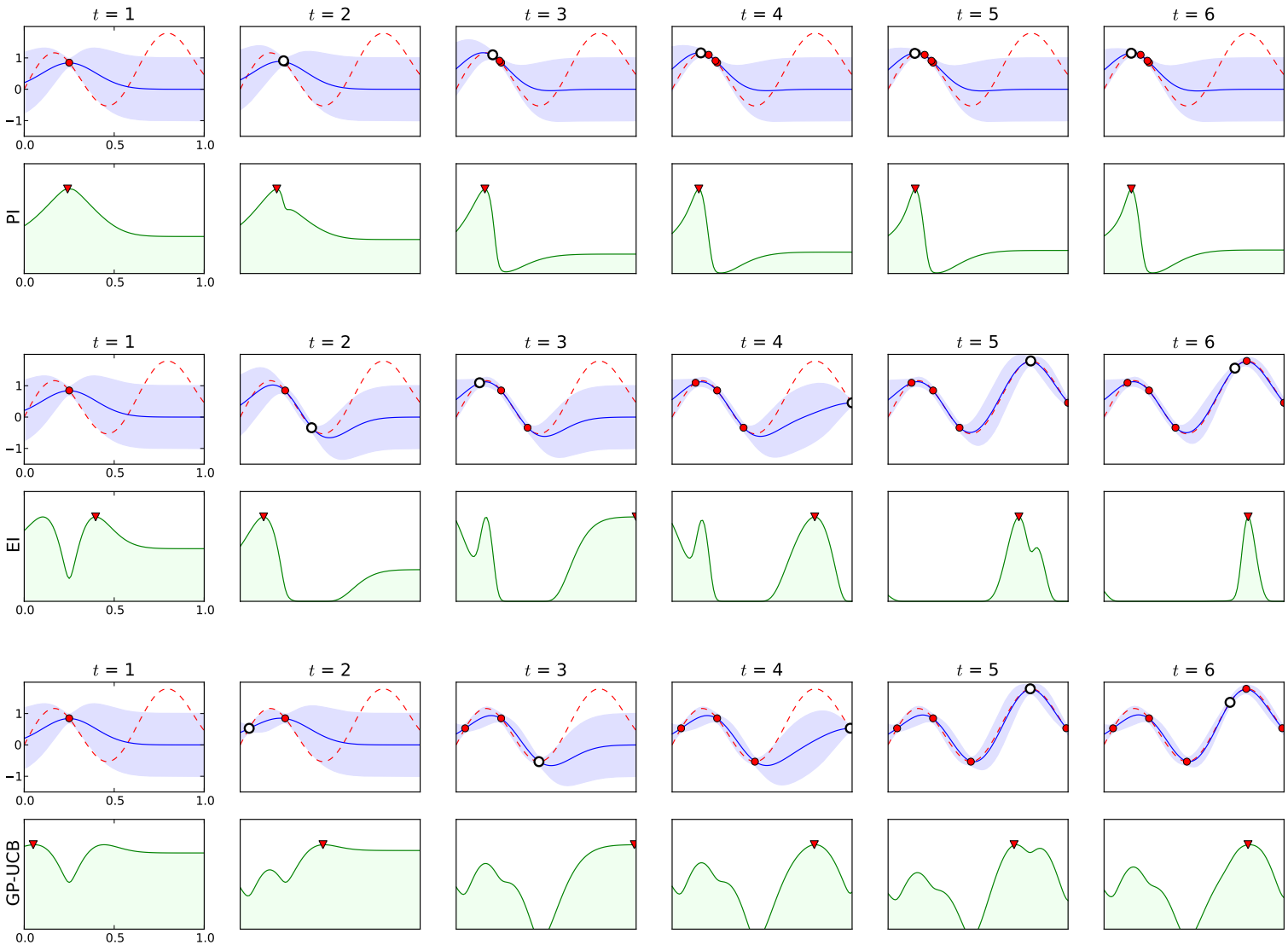
$$a_{EI}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) = \sigma\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)\left[\gamma(\mathbf{x})\Phi(\gamma(\mathbf{x})) + \mathcal{N}\left(\gamma(\mathbf{x}); 0, 1\right)\right]$$

- **Lower Confidence Bound** (LCB)

$$a_{EI}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) = \mu\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) - \kappa \cdot \sigma\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)$$

- Note: need to maximize the acquisition function to choose $\mathbf{x}_{next}$. Non-convex, but cheaper and simpler than original problem.

Acquisition functions from Brochu et al [1]



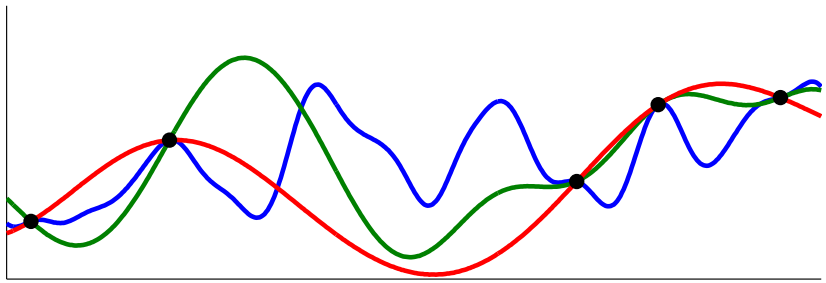
Marginalizing over hyperparameters

- Aquisition rules depends on GP hyperparameters (e.g. length scale)
- Snoek et al (NIPS, 2012 [2]) propose averaging acquisition functions over GP hyperparameters θ

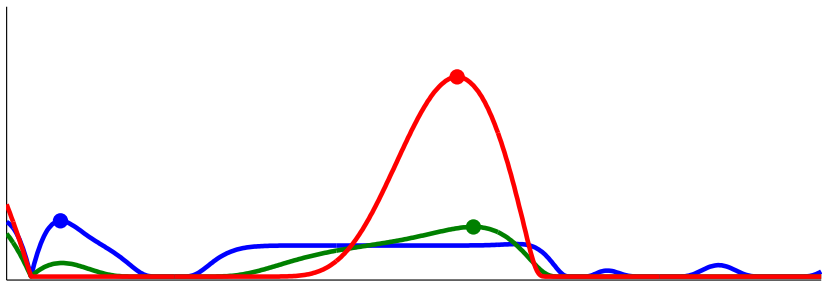
$$\hat{a}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}\right) = \int a\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) p\left(\theta | \{\mathbf{x}_n, y_n\}\right) d\theta$$

- Posterior $p\left(\theta | \{\mathbf{x}_n, y_n\}\right)$ can be efficiently computed by slice sampling.

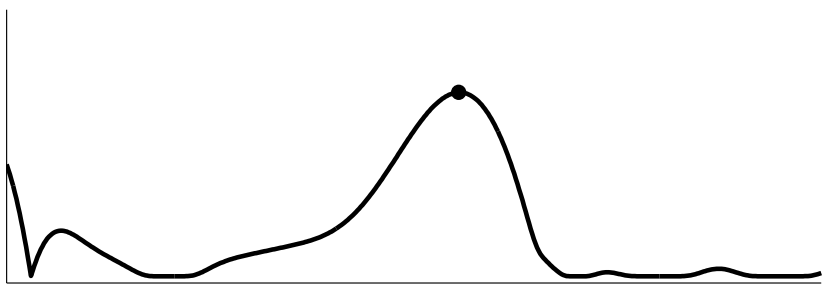
From Snoek et al (NIPS, 2012 [2])



(a) Posterior samples under varying hyperparameters

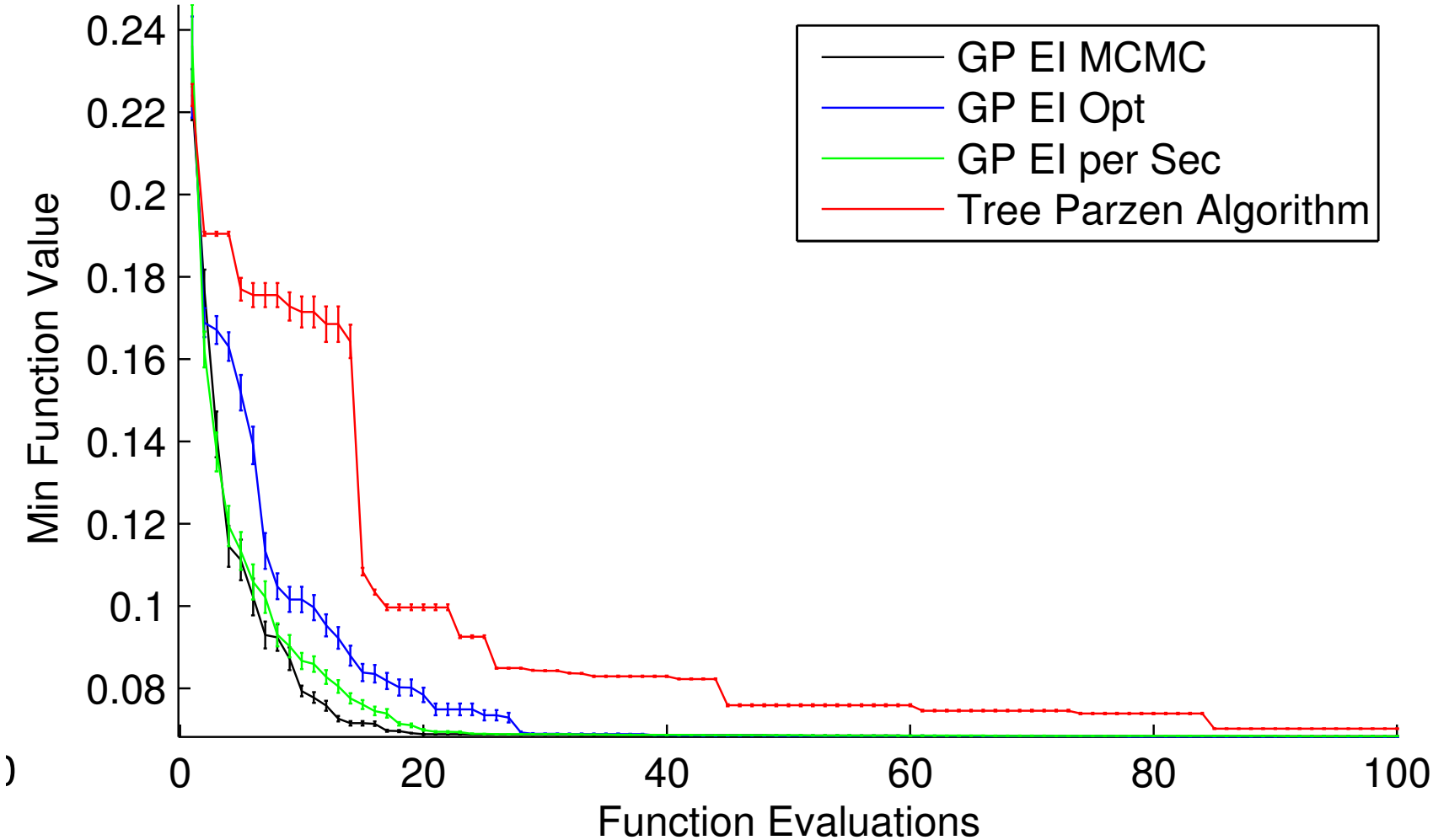


(b) Expected improvement under varying hyperparameters

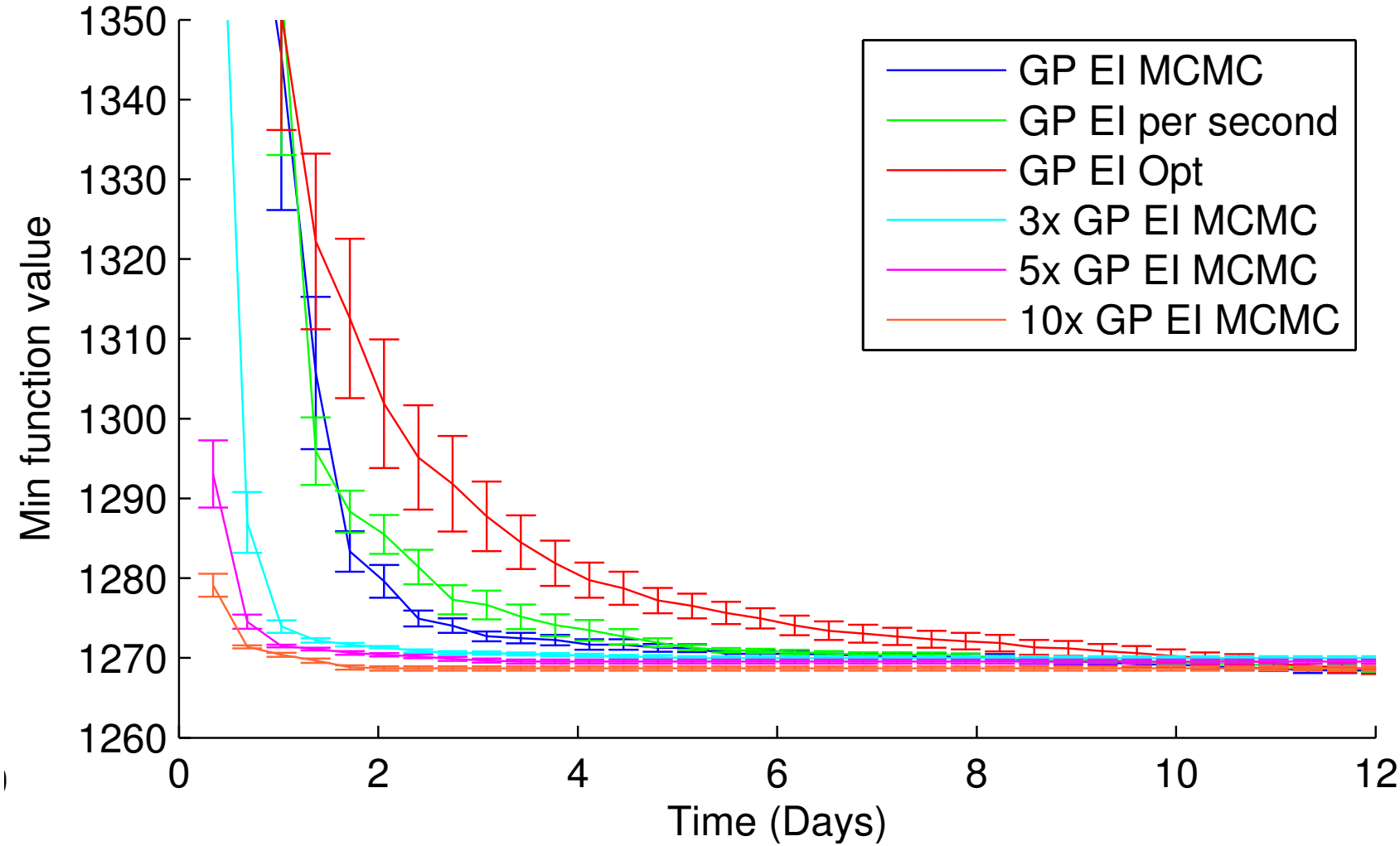


(c) Integrated expected improvement

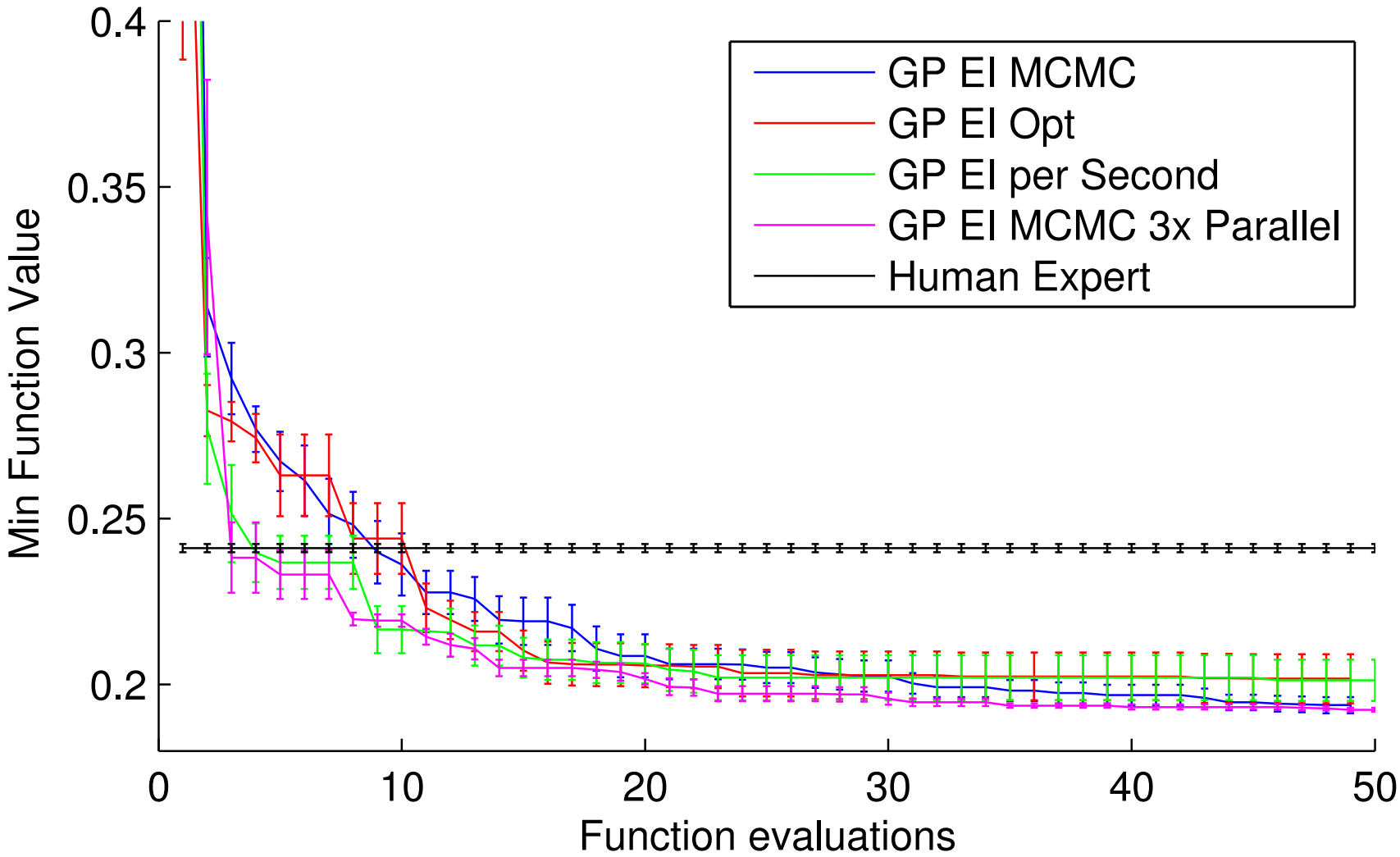
MNIST - Snoek et al (NIPS, 2012 [2])



Topic model - Snoek et al (NIPS, 2012 [2])



ConvNets - Snoek et al (NIPS, 2012 [2])



GPO in action: Intractable state-space [3]

- **State-space model** with **α-stable noise**

$$y_t \sim \text{AlphaStable}\left(y_t; \alpha, \exp(x_t)\right)$$
$$x_{t+1} = \mu + \phi\left(x_t - \mu\right) + \sigma_v \varepsilon_{t+1}$$

- Standard approach for parameter inference in SS models is PMMH.
- PDF does not exist in closed from for α-stable. Approximate Bayesian Computation (**SMC-ABC**).
- Posterior evaluations $\log \hat{p}(\theta_k | y_{1:T})$ are costly and noisy.
- **GPO** attractive as it uses few evaluations of the posterior.
- **GPO for normal (Laplace) approximation** of the posterior.
- **GPO is 60-100 times faster** than state-of-the-art PMMH.
- Application to 30-dim Gaussian copula with α-stable margins.

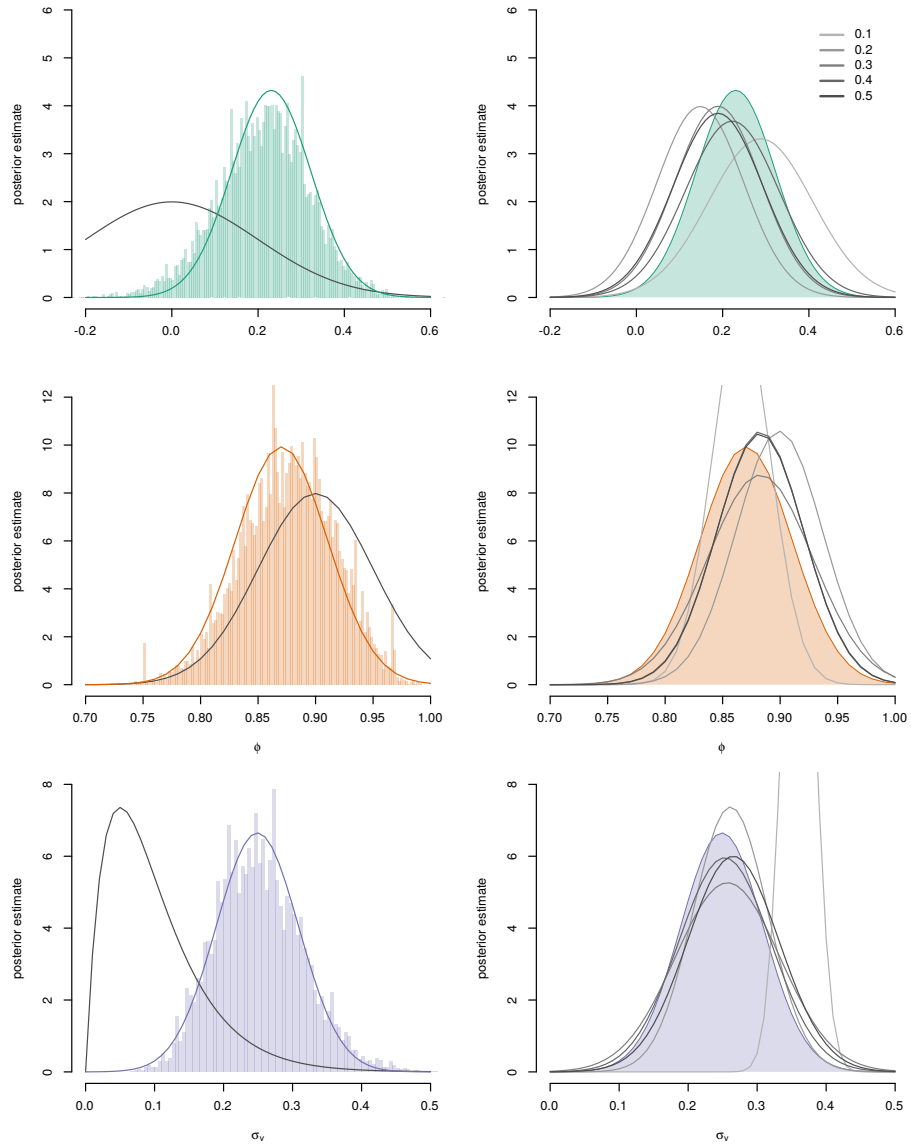
SMC-ABC-GPO

- SMC-ABC-GPO [3]:
  1. Compute an **estimate of the log posterior** at a parameter value θ_k, $z_k = \log \hat{p}(\theta_k | y_{1:T})$ using **SMC-ABC**.
  2. Update the **GP surrogate for the log posterior** using the available (**noisy**) evaluations $\{\theta_j, z_j\}_{j=1}^{k}$.
  3. Use the **acquisition rule** to determine the next evaluation point θ_{k+1}.

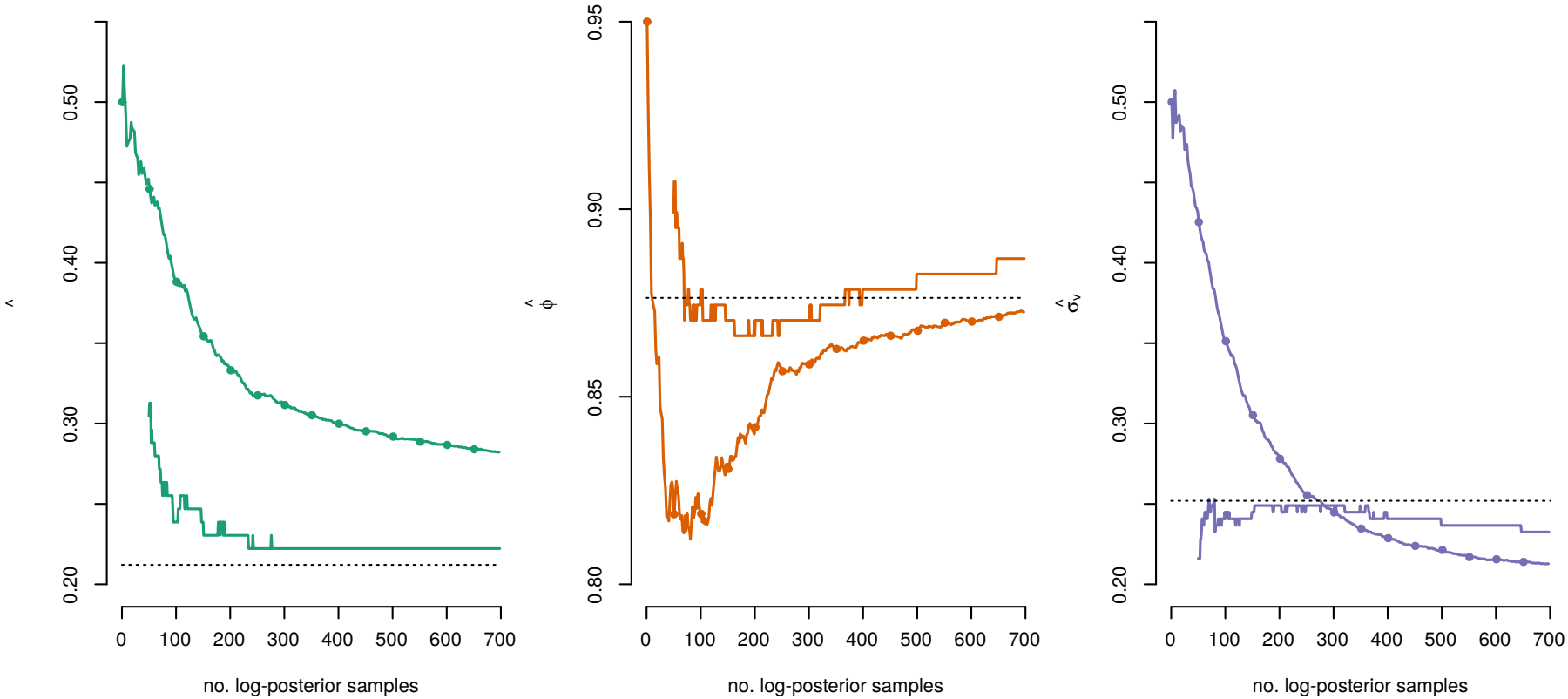
- End result from 1-3: smooth GP surrogate to the log posterior.

- Approximate **posterior covariance matrix** is obtained from finite differences of the GP posterior mean function.

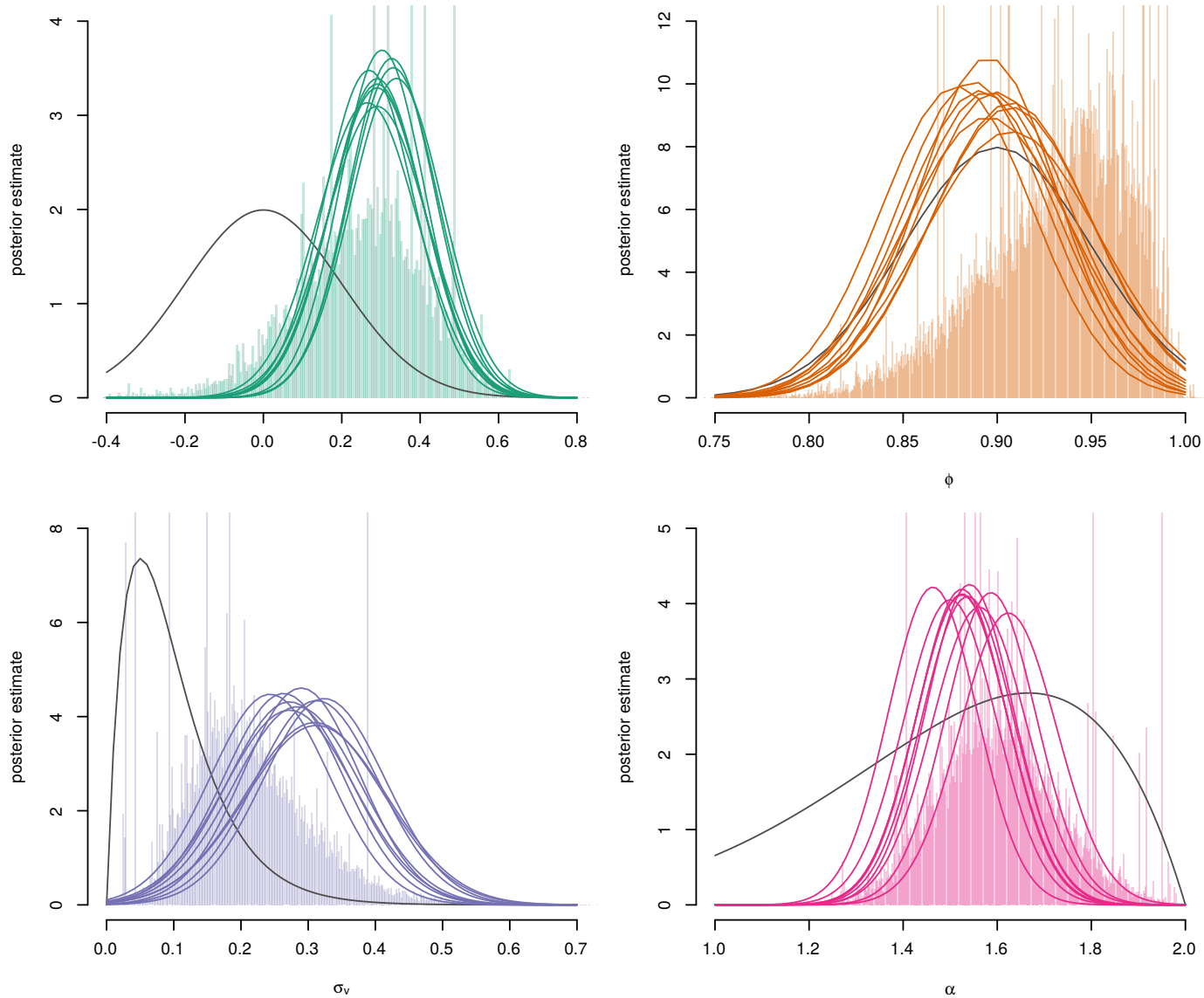
Sanity check: Linear Gaussian State Space



Sanity check: Linear Gaussian State Space



Analysis of returns from coffee futures




E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.


J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959, 2012.


J. Dahlin, M. Villani, and T. B. Schön, "Bayesian optimisation for approximate inference in state-space models with intractable likelihoods," *arXiv preprint arXiv:1506.06975*, 2015.