Gaussian Processes

Dan Cervone

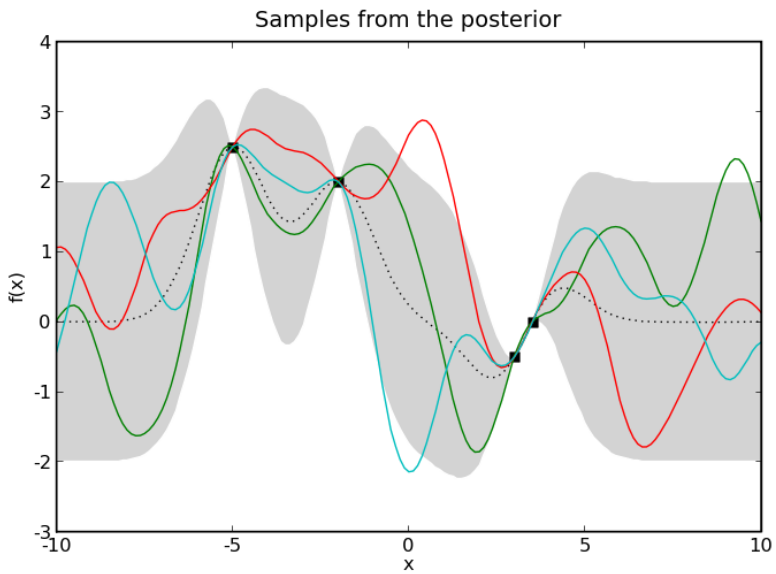
NYU CDS

November 10, 2015

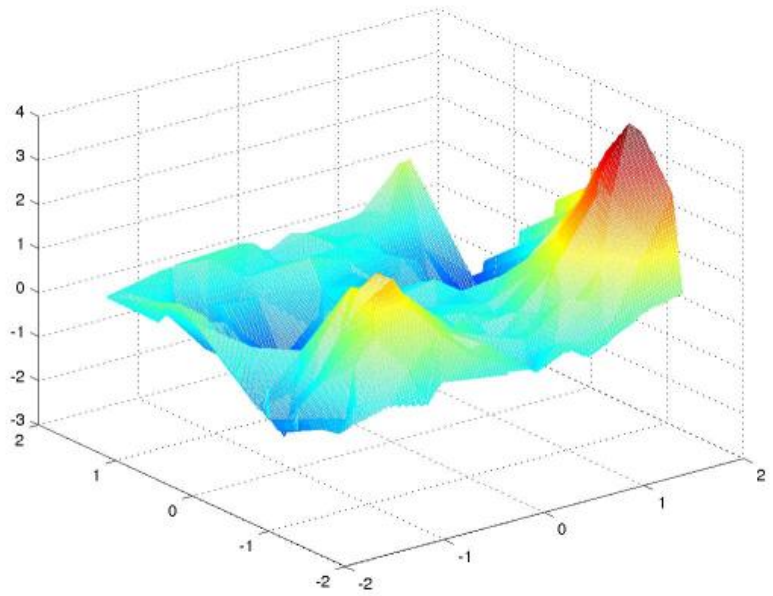
What are Gaussian processes?

GPs let us do Bayesian inference on *functions*. Using GPs we can:

- Interpolate spatial data
- Forecast time series
- Represent latent surfaces for classification, point processes, etc.
- Emulate likelihoods and complex, black-box functions
- Model cool stuff across many scientific disciplines!



[https://pythonhosted.org/infpy/gps.html]



[http://becs.aalto.fi/en/research/bayes/mcmcstuff/traindata.jpg]

Preliminaries

The basic setup:

- Data set $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Inputs $\mathbf{x}_i \in \mathbb{S} \subset \mathbb{R}^D$.
- Outputs $y_i \in \mathbb{R}$.

$$
\begin{aligned}
x_i &\sim p(x) \\
y_i &= f(x_i) + \epsilon_i \\
\epsilon_i &\overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)
\end{aligned}
$$

Preliminaries

The basic setup:

- Data set $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Inputs $\mathbf{x}_i \in \mathbb{S} \subset \mathbb{R}^D$.
- Outputs $y_i \in \mathbb{R}$.

$$
\begin{aligned}
x_i &\sim p(x) \\
y_i &= f(x_i) + \epsilon_i \\
\epsilon_i &\overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)
\end{aligned}
$$

Definition

f is a Gaussian process if for any collection $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{S}, i = 1, \ldots, n\}$,

$$
\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{pmatrix} \sim \mathcal{N}(\mu(\mathbf{X}), K(\mathbf{X}, \mathbf{X}))
$$

Mean, covariance functions

GPs characterized by mean, covariance functions:

- Mean function: $\mu(\mathbf{x})$.
- WLOG, we can assume $\mu = 0$. (Why?)
- Covariance function k where

$$[K(\mathbf{X}, \mathbf{X})]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \text{Cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)).$$

Mean, covariance functions

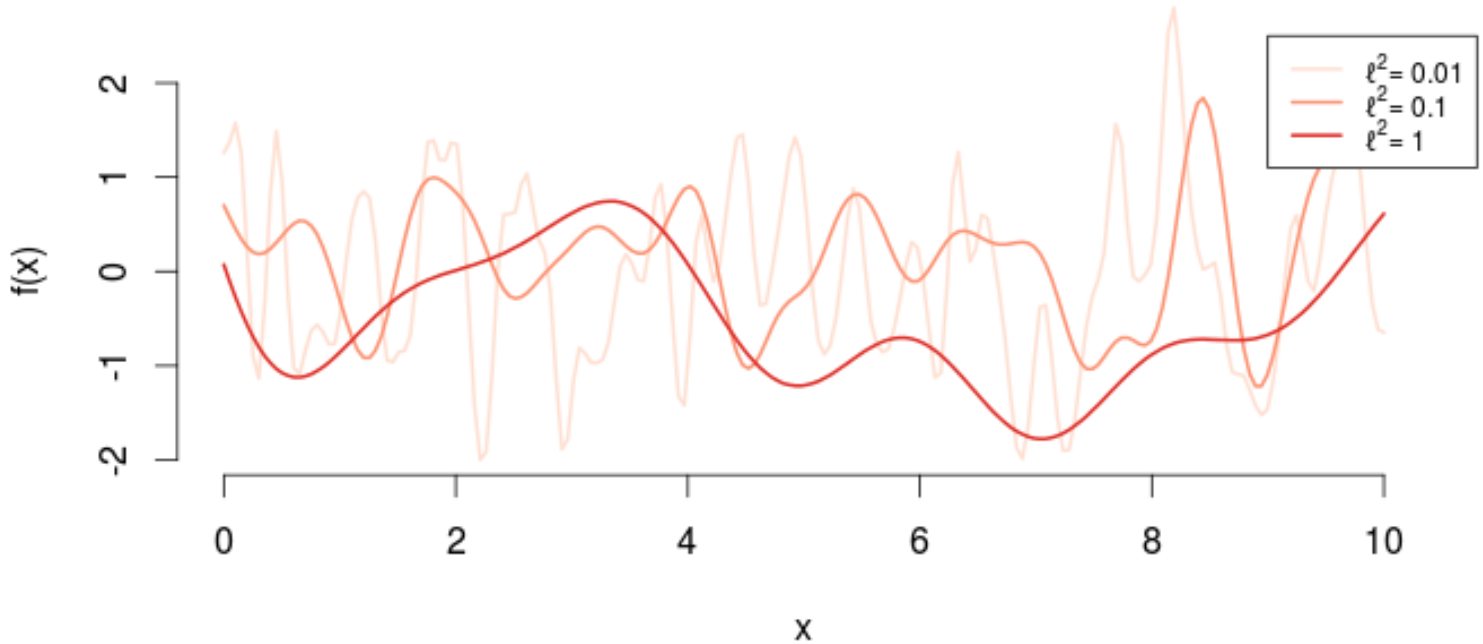
GPs characterized by mean, covariance functions:

- Mean function: $\mu(\mathbf{x})$.
- WLOG, we can assume $\mu = 0$. (Why?)
- Covariance function k where

$$[K(\mathbf{X}, \mathbf{X})]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \text{Cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)).$$

Example:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\ell^2}\right) \text{ (squared exponential)}$$



GP regression (prediction)

Interpolation/prediction at target locations:

- *(Noise-free observations)* Observe $\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \ldots, n\}$.
- *(Noisy observations)* Observe $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Want to predict $\mathbf{f}^* = \{f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_k^*)\}$ at $\mathbf{x}^*$.

GP regression (prediction)

Interpolation/prediction at target locations:

- *(Noise-free observations)* Observe $\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \ldots, n\}$.
- *(Noisy observations)* Observe $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Want to predict $\mathbf{f}^* = \{f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_k^*)\}$ at $\mathbf{X}^*$.

$$\left.\begin{aligned} \begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} | \mathbf{X}, \mathbf{X}^* &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix}\right) \\ \mathbf{f}^* | \mathbf{f}, \mathbf{X}, \mathbf{X}^* &\sim \mathcal{N}\Big(K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X})]^{-1}\mathbf{f}, \\ &\qquad K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X})]^{-1}K(\mathbf{X}, \mathbf{X}^*)\Big) \end{aligned}\right\} \boxed{\text{Prediction with noise-free data}}$$

GP regression (prediction)

Interpolation/prediction at target locations:

- *(Noise-free observations)* Observe $\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \ldots, n\}$.
- *(Noisy observations)* Observe $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Want to predict $\mathbf{f}^* = \{f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_k^*)\}$ at $\mathbf{x}^*$.

$$\left.\begin{aligned} \begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} | \mathbf{X}, \mathbf{X}^* &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix}\right) \\ \mathbf{f}^* | \mathbf{f}, \mathbf{X}, \mathbf{X}^* &\sim \mathcal{N}\Big(K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X})]^{-1}\mathbf{f}, \\ &\qquad K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X})]^{-1}K(\mathbf{X}, \mathbf{X}^*)\Big) \end{aligned}\right\} \text{Prediction with noise-free data}$$

$$\left.\begin{aligned} \begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} | \mathbf{X}, \mathbf{X}^* &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}_n & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix}\right) \\ \mathbf{f}^* | \mathbf{y}, \mathbf{X}, \mathbf{X}^* &\sim \mathcal{N}\Big(K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}_n]^{-1}\mathbf{y}, \\ &\qquad K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}_n]^{-1}K(\mathbf{X}, \mathbf{X}^*)\Big) \end{aligned}\right\} \text{Prediction with noisy data}$$

GP regression (prediction)

Some cool things we've noticed:

- $\mathbf{f}, \mathbf{f}^*, \mathbf{y}, \mathbf{y}^*$ are all jointly Gaussian.
- GP regression gives us interval (distributional) predictions for free.

Prediction using noise-free vs. noisy data:

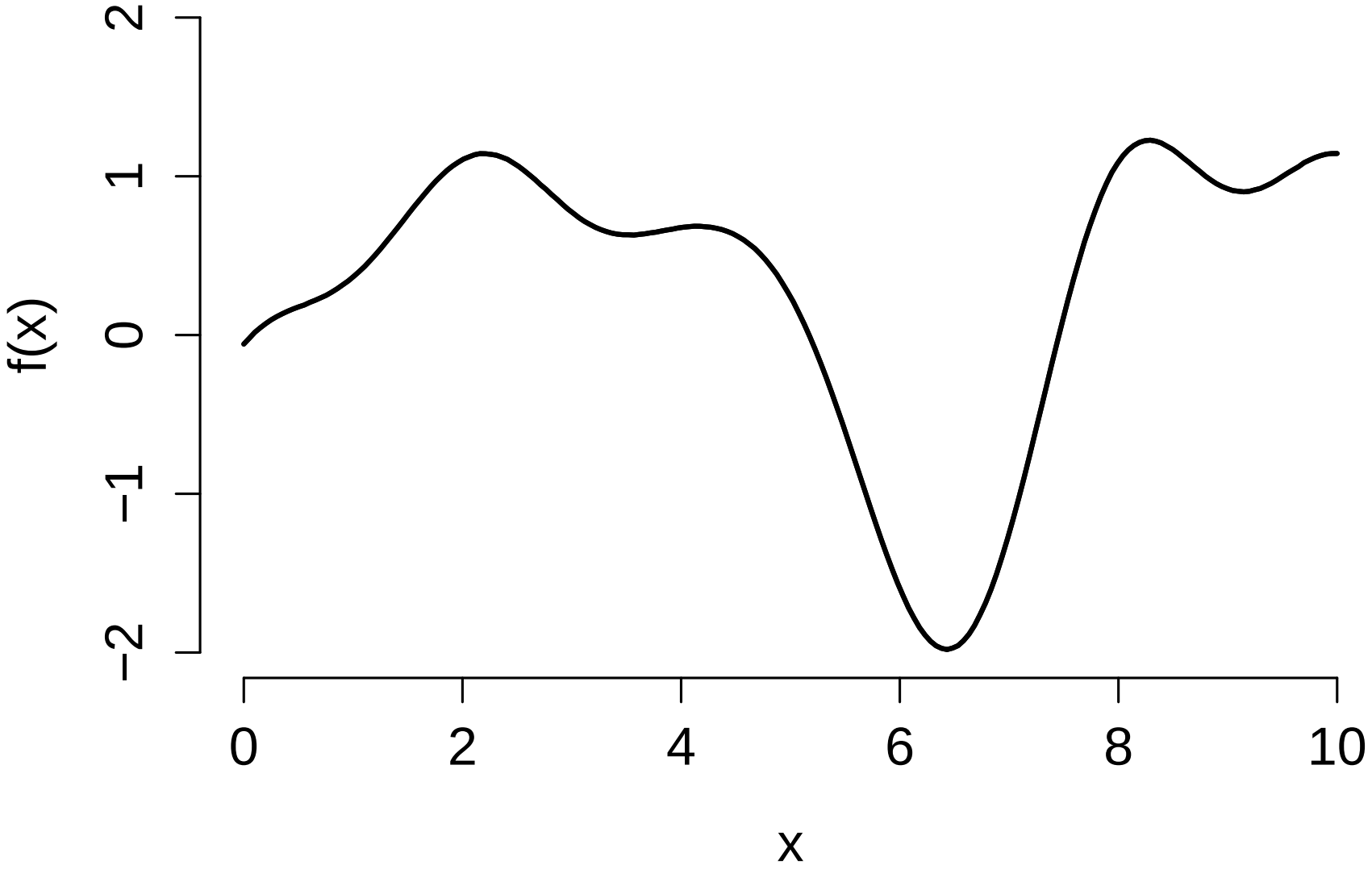
- Which situation is more likely in practice?

The "nugget" $\sigma_\epsilon^2 \mathbf{I}_n$:

- Arises due to measurement error or high-frequency behavior.
- Provides numerical stability and regularization.

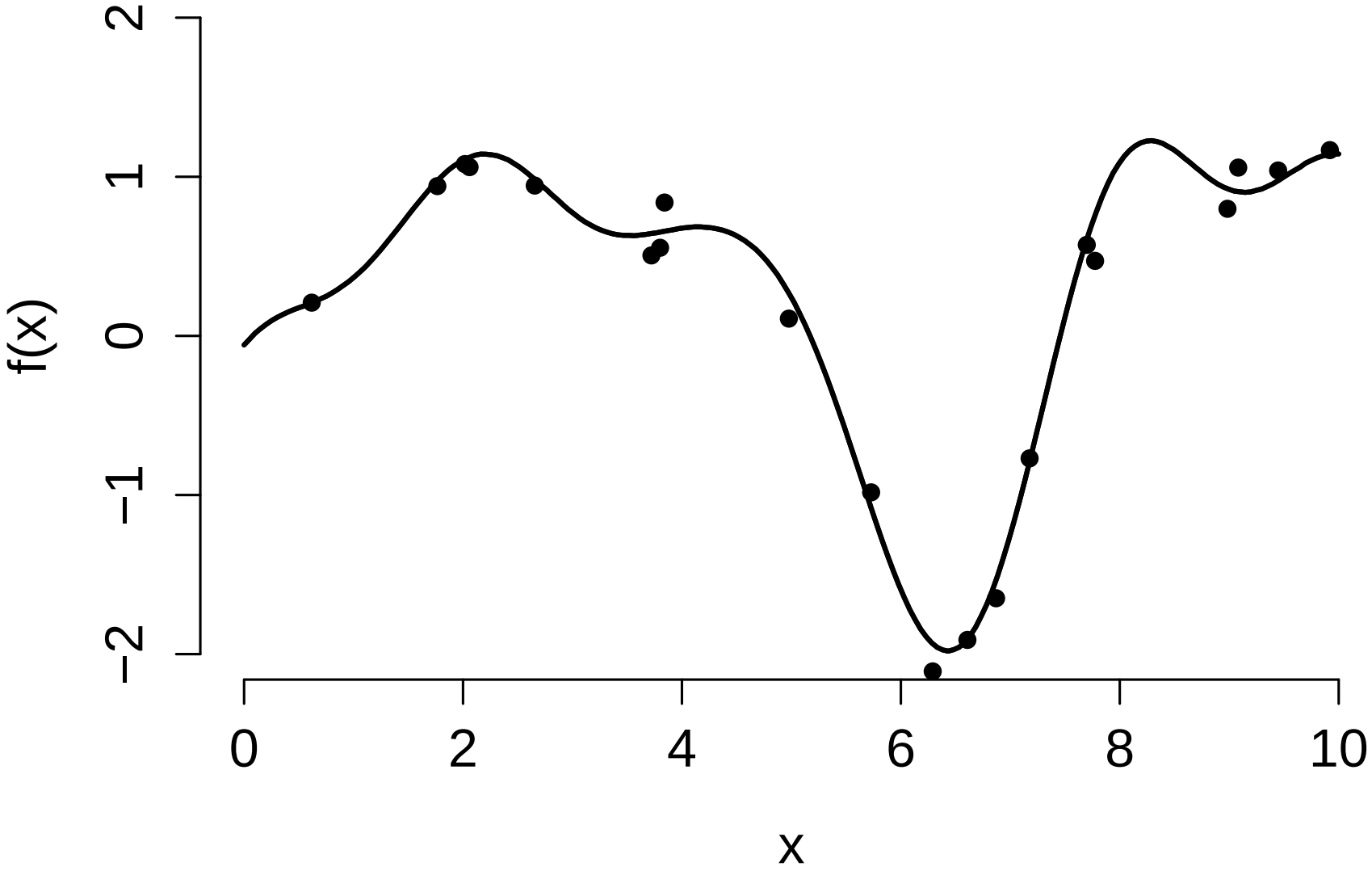
Illustrating GP regression

TRUTH: $\tau^2 = 1, \ell^2 = 1, \sigma_\epsilon^2 = 0.01$.

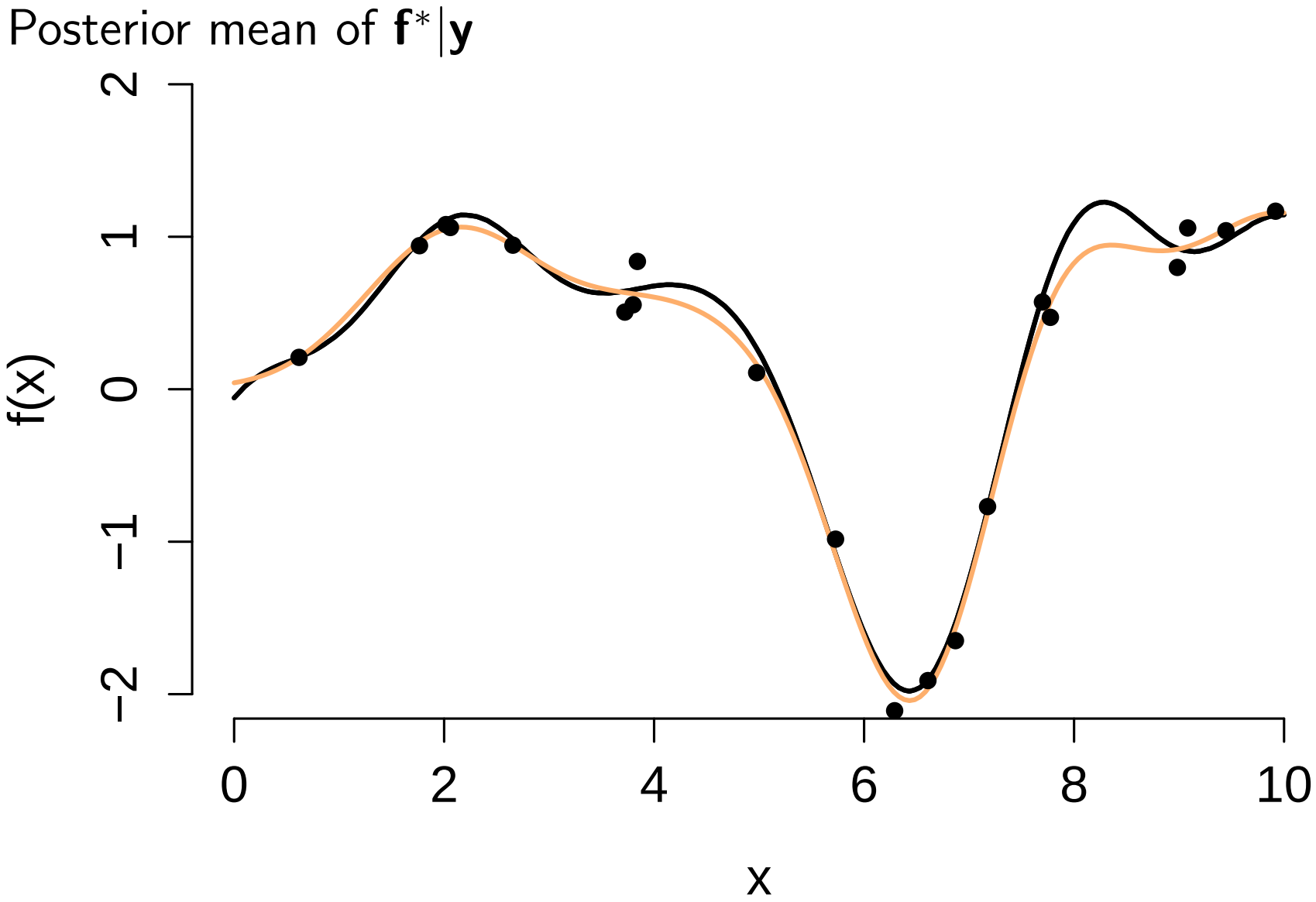


Illustrating GP regression

Sample $\{(x_i, y_i), i = 1, \ldots 20\}$

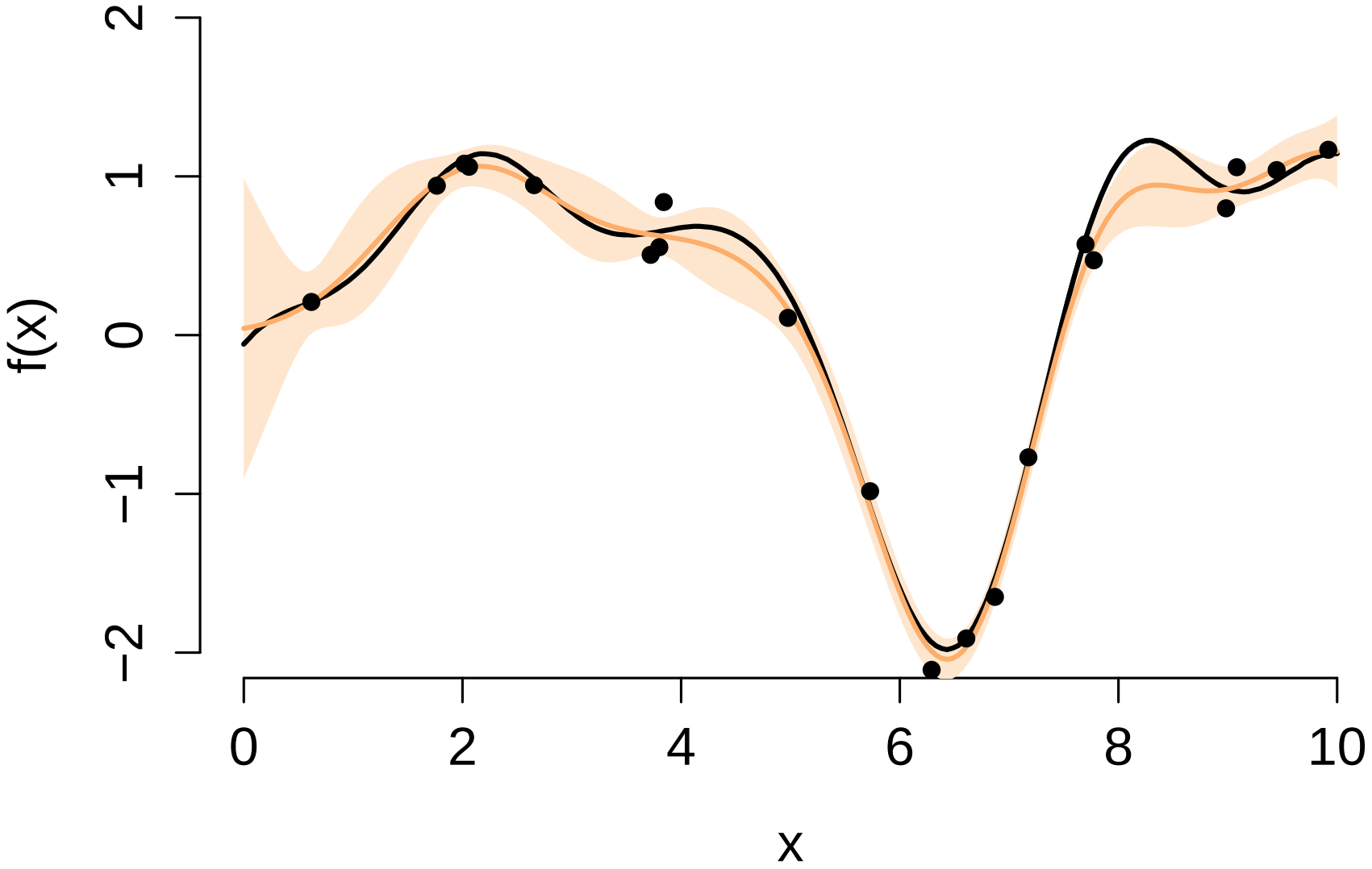


Illustrating GP regression



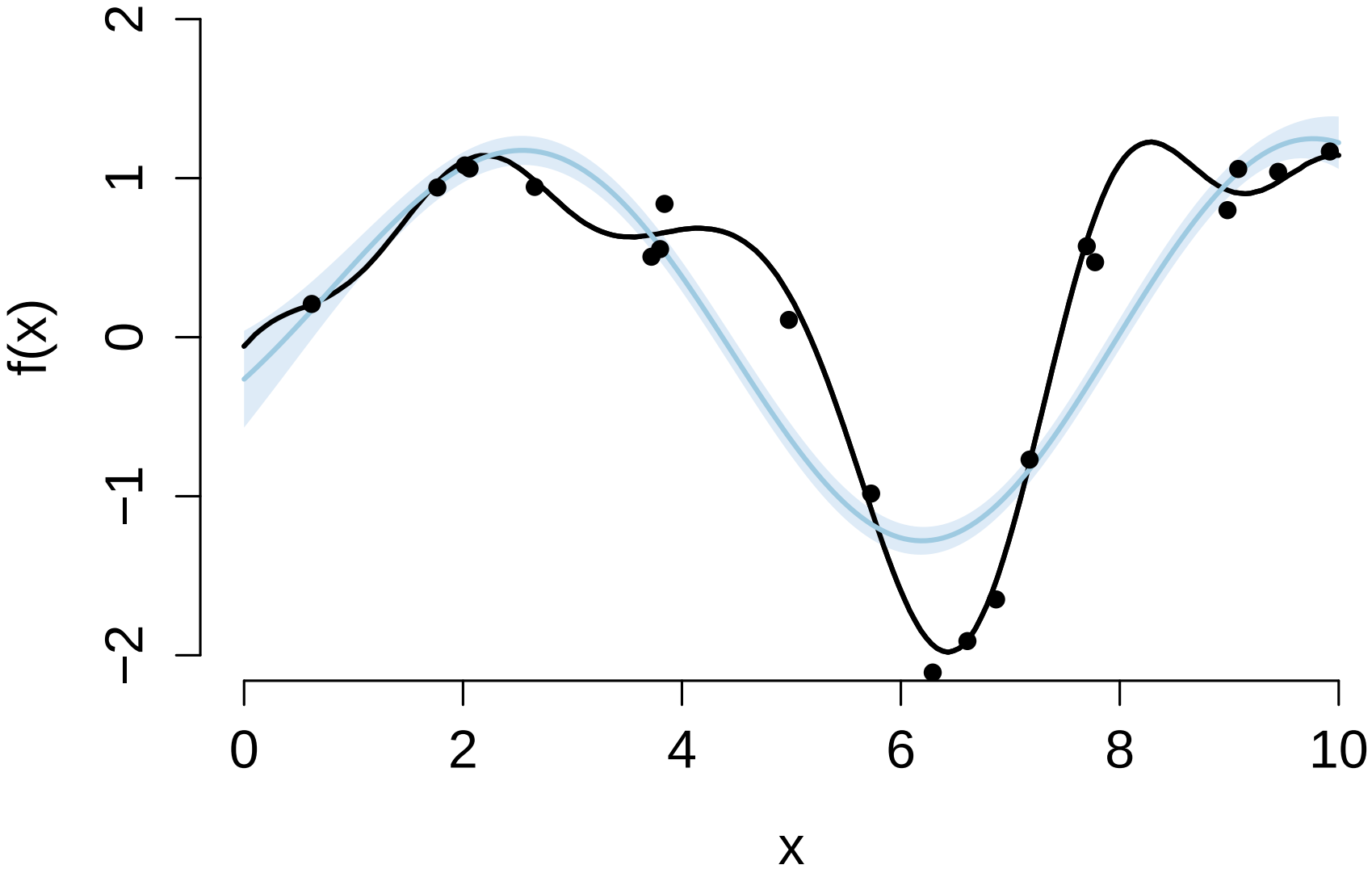
Illustrating GP regression

95% prediction interval for $\mathbf{f}^*|\mathbf{y}$



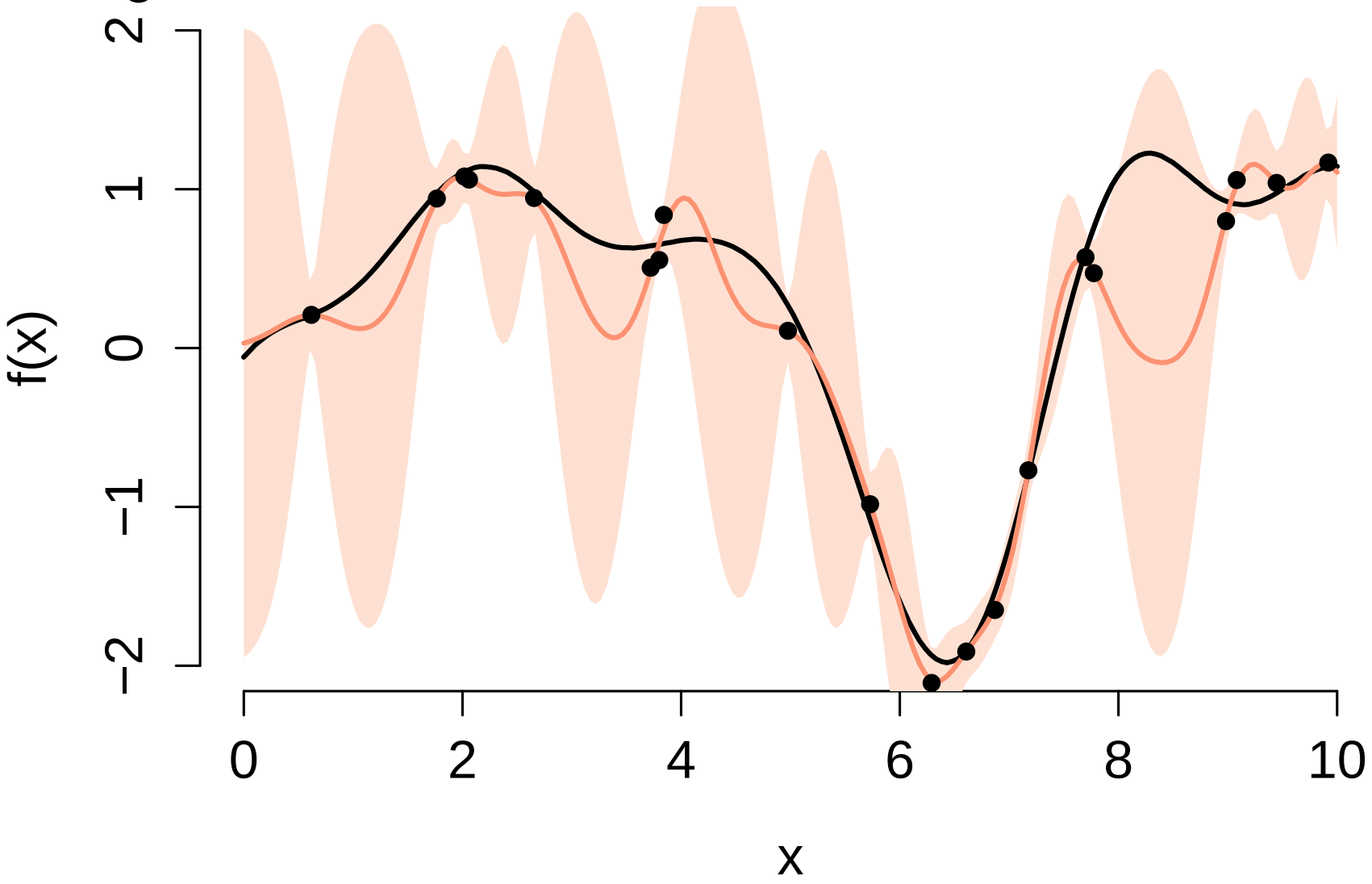
Illustrating GP regression

Fitting GP with $\ell^2 = 10$:



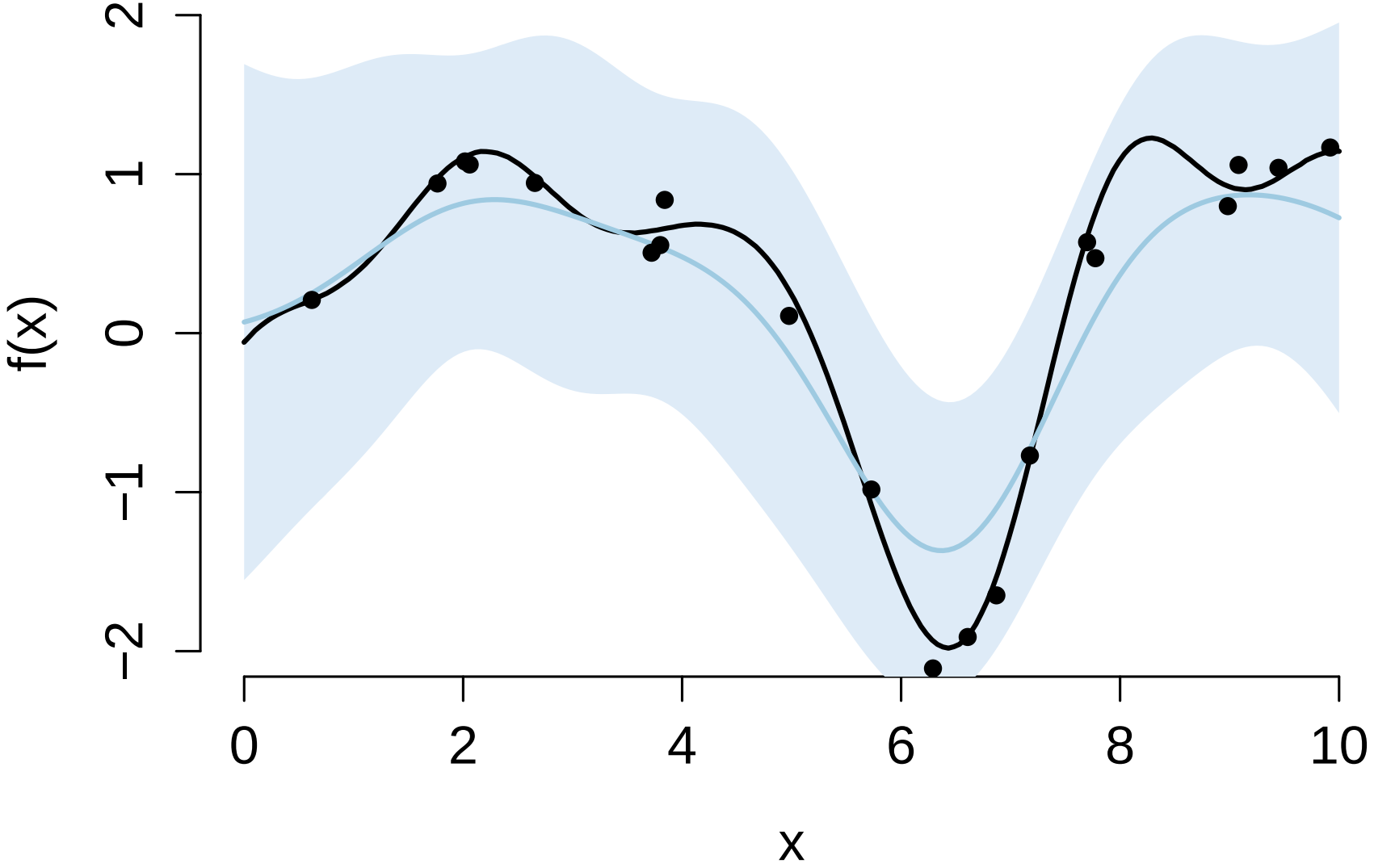
Illustrating GP regression

Fitting GP with $\ell^2 = 0.1$:



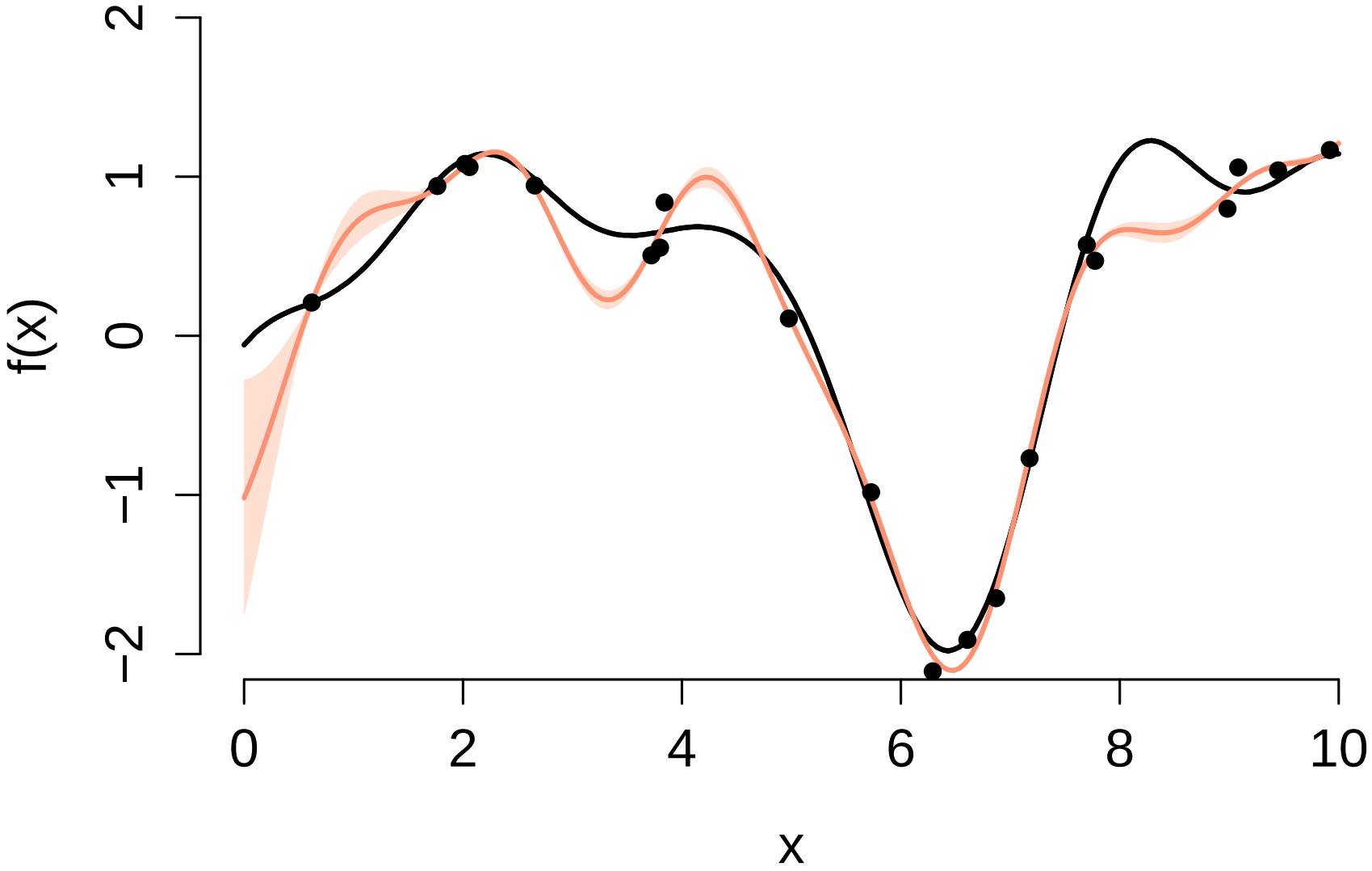
Illustrating GP regression

Fitting GP with $\sigma_\epsilon^2 = 1$:



Illustrating GP regression

Fitting GP with $\sigma_\epsilon^2 = 0.0001$:



GPs and Bayesian linear regression

Assume $f(\mathbf{x}_i)$ is linear in p-dimensional feature vector of $\mathbf{x}_i$:

$$\begin{aligned} f(\mathbf{x}_i) &= \phi(\mathbf{x}_i)'\mathbf{w} \\ &= \boldsymbol{\phi}_i'\mathbf{w} \end{aligned}$$

GPs and Bayesian linear regression

Assume $f(\mathbf{x}_i)$ is linear in p-dimensional feature vector of $\mathbf{x}_i$:

$$\begin{aligned} f(\mathbf{x}_i) &= \phi(\mathbf{x}_i)'\mathbf{w} \\ &= \phi_i'\mathbf{w} \end{aligned}$$

Usual Bayesian regression setup for ϕ:

$$\begin{aligned} y_i|\mathbf{X} &\overset{ind}{\sim} \mathcal{N}(\phi_i'\mathbf{w}, \sigma_\epsilon^2) && \text{(likelihood)} \\ \mathbf{w} &\sim \mathcal{N}(0, \Sigma) && \text{(prior)} \\ \mathbf{w}|\mathbf{y}, \mathbf{X} &\sim \mathcal{N}(\hat{\mathbf{w}}, A^{-1}) && \text{(posterior)} \\ f^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^* &\sim \mathcal{N}((\phi^*)'\hat{\mathbf{w}}, (\phi^*)'A^{-1}\phi^*) && \text{(posterior predictive)} \end{aligned}$$

where

- $\hat{\mathbf{w}} = A^{-1}\mathbf{\Phi}\mathbf{y}/\sigma_\epsilon^2$.
- $A = \mathbf{\Phi}\mathbf{\Phi}'/\sigma_\epsilon^2 + \Sigma^{-1}$.
- $\mathbf{\Phi} = p \times n$ matrix stacking $\phi_i, i = 1, \ldots, n$ columnwise.

GPs and Bayesian linear regression

After some matrix algebra (Woodbury identity!), we can write this as:

$$f^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^* \sim \mathcal{N}\Big((\phi^*)'\Sigma\mathbf{\Phi}[\mathbf{\Phi}'\Sigma\mathbf{\Phi} + \sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{y}, \\ (\phi^*)'\Sigma\phi^* - (\phi^*)'\Sigma\mathbf{\Phi}[\mathbf{\Phi}'\Sigma\mathbf{\Phi} + \sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{\Phi}'\Sigma\phi^*\Big)$$

- Taking $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)'\Sigma\phi(\mathbf{x}_j)$, we get familiar GP prediction expression.
- Thus $\{\text{Bayesian regression}\} \subset \{\text{Gaussian processes}\}$.
- $\{\text{Gaussian processes}\} \subset \{\text{Bayesian regression}\}$?

GPs and Bayesian linear regression

After some matrix algebra (Woodbury identity!), we can write this as:

$$f^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^* \sim \mathcal{N}\Big((\phi^*)'\Sigma\mathbf{\Phi}[\mathbf{\Phi}'\Sigma\mathbf{\Phi} + \sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{y}, \\ (\phi^*)'\Sigma\phi^* - (\phi^*)'\Sigma\mathbf{\Phi}[\mathbf{\Phi}'\Sigma\mathbf{\Phi} + \sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{\Phi}'\Sigma\phi^*\Big)$$

- Taking $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)'\Sigma\phi(\mathbf{x}_j)$, we get familiar GP prediction expression.
- Thus $\{\text{Bayesian regression}\} \subset \{\text{Gaussian processes}\}$.
- $\{\text{Gaussian processes}\} \subset \{\text{Bayesian regression}\}$?

"Kernel trick": feature vectors ϕ only enter as inner products $\mathbf{\Phi}'\Sigma\mathbf{\Phi}$, $(\phi^*)'\Sigma\mathbf{\Phi}$, or $(\phi^*)'\Sigma\phi^*$.

- Kernel (covariance function) $k(\cdot, \cdot)$ spares us from ever calculating $\phi(\mathbf{x})$.
- Where have we seen this before?

Covariance functions

Common choices:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{2\ell}\right) \text{ (exponential)}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\ell^2}\right) \text{ (squared exponential)}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \left(1 - \frac{3||\mathbf{x}_i - \mathbf{x}_j||}{2\theta} + \frac{||\mathbf{x}_i - \mathbf{x}_j||^3}{2\theta^3}\right) \mathbf{1}[||\mathbf{x}_i - \mathbf{x}_j|| \leq \theta] \text{ (spherical)}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{\tau^2}{\Gamma(\nu)} \left(\frac{||\mathbf{x}_i - \mathbf{x}_j||}{2\phi}\right)^{\nu} B_\nu(\phi||\mathbf{x}_i - \mathbf{x}_j||) \text{ (matérn)}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 + \tau^2(\mathbf{x}_i - \mathbf{c})'(\mathbf{x}_j - \mathbf{c}) \text{ (linear)}$$

Covariance functions

Properties

Isotrophy (stationarity)

- Covariance only depends on distance: $k(\mathbf{x}_i, \mathbf{x}_j) = c(||\mathbf{x}_i - \mathbf{x}_j||)$.
- Common in many GP applications.

Covariance functions

Properties

Isotrophy (stationarity)

- Covariance only depends on distance: $k(\mathbf{x}_i, \mathbf{x}_j) = c(||\mathbf{x}_i - \mathbf{x}_j||)$.
- Common in many GP applications.

Differentiability

- Sample paths $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$ may be m times differentiable.
- Example of non-differentiable Gaussian Process?

Covariance functions

Properties

Isotrophy (stationarity)

- Covariance only depends on distance: $k(\mathbf{x}_i, \mathbf{x}_j) = c(||\mathbf{x}_i - \mathbf{x}_j||)$.
- Common in many GP applications.

Differentiability

- Sample paths $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$ may be m times differentiable.
- Example of non-differentiable Gaussian Process?

Compact support

- For any $\mathbf{x}_1, \{\mathbf{x}_2 : k(\mathbf{x}_1, \mathbf{x}_2) \neq 0\}$ is compact.
- Provides sparsity in covariance matrix.

Covariance functions

Properties

Isotrophy (stationarity)

- Covariance only depends on distance: $k(\mathbf{x}_i, \mathbf{x}_j) = c(||\mathbf{x}_i - \mathbf{x}_j||)$.
- Common in many GP applications.

Differentiability

- Sample paths $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$ may be m times differentiable.
- Example of non-differentiable Gaussian Process?

Compact support

- For any $\mathbf{x}_1, \{\mathbf{x}_2 : k(\mathbf{x}_1, \mathbf{x}_2) \neq 0\}$ is compact.
- Provides sparsity in covariance matrix.

Combining covariance functions

- Assume k_1 and k_2 are valid covariance functions:
- $k = k_1 + k_2$ is a valid covariance function.
- $k = k_1 \times k_2$ is a valid covariance function.
- $k_g = k(g(\mathbf{x}_1), g(\mathbf{x}_2))$ is a valid covariance function.

Covariance functions

Properties

Cov. Function	Isotropic	Times differentiable	Compact
Exponential	Yes	0	No
Squared exponential	Yes	∞	No
Spherical	Yes	0	Yes
Matérn	Yes	ν	No
Linear	No	∞	No

Parameter estimation and inference

Marginal likelihood:

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

$$\mathbf{y}|\theta \sim \mathcal{N}(0, K_\theta(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I})$$

Parameter estimation and inference

Marginal likelihood:

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

$$\mathbf{y}|\theta \sim \mathcal{N}(0, K_\theta(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I})$$

Thus

$$\log(p(\mathbf{y}|\theta)) = -\frac{1}{2}\mathbf{y}'\mathbf{K}_y\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| + c$$

$$\frac{\partial}{\partial\theta_j}\log(p(\mathbf{y}|\theta)) = \frac{1}{2}\mathbf{y}'\mathbf{K}_y^{-1}\left(\frac{\partial}{\partial\theta_j}\mathbf{K}_y\right)\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\text{tr}\left(\mathbf{K}_y^{-1}\frac{\partial}{\partial\theta_j}\mathbf{K}_y\right)$$

where $\mathbf{K}_y = K_\theta(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}$.

Parameter estimation and inference

Marginal likelihood:

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

$$\mathbf{y}|\theta \sim \mathcal{N}(0, K_\theta(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I})$$

Thus

$$\log(p(\mathbf{y}|\theta)) = -\frac{1}{2}\mathbf{y}'\mathbf{K}_y\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| + c$$

$$\frac{\partial}{\partial\theta_j}\log(p(\mathbf{y}|\theta)) = \frac{1}{2}\mathbf{y}'\mathbf{K}_y^{-1}\left(\frac{\partial}{\partial\theta_j}\mathbf{K}_y\right)\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\text{tr}\left(\mathbf{K}_y^{-1}\frac{\partial}{\partial\theta_j}\mathbf{K}_y\right)$$

where $\mathbf{K}_y = K_\theta(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}$.

- Can use any gradient-based method to maximize (log) marginal likelihood.
- Non-convex, so typically multiple solutions exist.
- Can also be fully Bayesian: supply prior $p(\theta)$ and sample posterior $p(\theta|\mathbf{y})$.

Latent GPs

We can generalize the observed data process $y_i = f(\mathbf{x}_i) + \epsilon_i$ by writing:

$$y_i \sim p(y|f(\mathbf{x}_i)).$$

For example:

Latent GPs

We can generalize the observed data process $y_i = f(\mathbf{x}_i) + \epsilon_i$ by writing:

$$y_i \sim p(y|f(\mathbf{x}_i)).$$

For example:

- Binary classification

$$P(y_i = 1|f(\mathbf{x}_i)) = \sigma(f(\mathbf{x}_i))$$

Latent GPs

We can generalize the observed data process $y_i = f(\mathbf{x}_i) + \epsilon_i$ by writing:

$$y_i \sim p(y|f(\mathbf{x}_i)).$$

For example:

- Binary classification

$$P(y_i = 1|f(\mathbf{x}_i)) = \sigma(f(\mathbf{x}_i))$$

- k-class classification

$$P(y_i = c_j|f_1(\mathbf{x}_i), \ldots, f_k(\mathbf{x}_i)) = \frac{\exp(f_j(\mathbf{x}_i))}{\sum_{j'=1}^{k} \exp(f_{j'}(\mathbf{x}_i))}$$

Latent GPs

We can generalize the observed data process $y_i = f(\mathbf{x}_i) + \epsilon_i$ by writing:

$$y_i \sim p(y|f(\mathbf{x}_i)).$$

For example:

- Binary classification

$$P(y_i = 1|f(\mathbf{x}_i)) = \sigma(f(\mathbf{x}_i))$$

- k-class classification

$$P(y_i = c_j|f_1(\mathbf{x}_i), \ldots, f_k(\mathbf{x}_i)) = \frac{\exp(f_j(\mathbf{x}_i))}{\sum_{j'=1}^{k} \exp(f_{j'}(\mathbf{x}_i))}$$

- Inhomogeneous Poisson process

$$y(t) \sim \mathcal{PP}(\lambda(t))$$
$$\log(\lambda(t)) = f(t)$$

Example: Binary GP classification

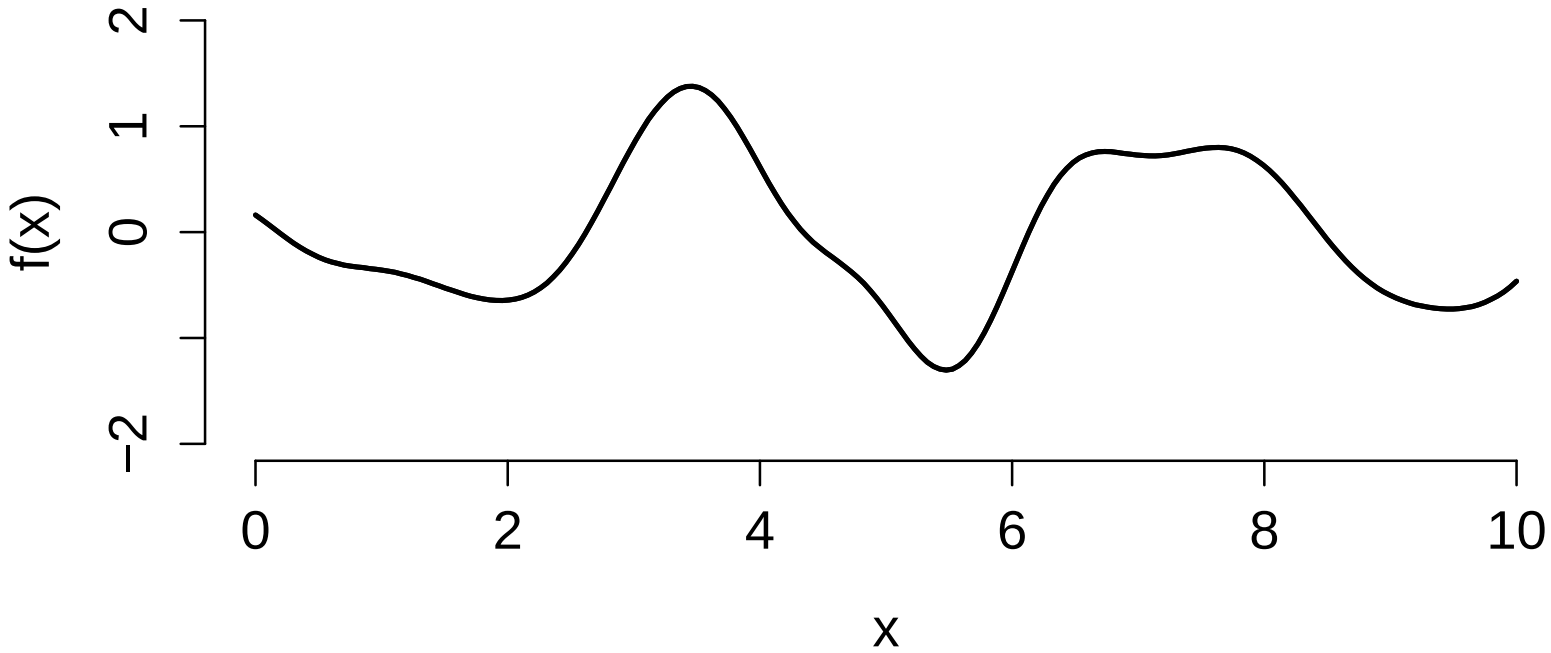
$$
\begin{aligned}
x_i &\overset{iid}{\sim} \text{Unif}(0, 10) \\
f(x) &\sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ with } k(x_i, x_j) = \exp(-(x_i - x_j)^2/4) \\
y_i &\overset{ind}{\sim} \text{Bern}\left(\frac{1}{1 + \exp(-f(x_i))}\right)
\end{aligned}
$$

Example: Binary GP classification

$$x_i \overset{iid}{\sim} \text{Unif}(0, 10)$$

$$f(x) \sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ with } k(x_i, x_j) = \exp(-(x_i - x_j)^2/4)$$

$$y_i \overset{ind}{\sim} \text{Bern}\left(\frac{1}{1 + \exp(-f(x_i))}\right)$$

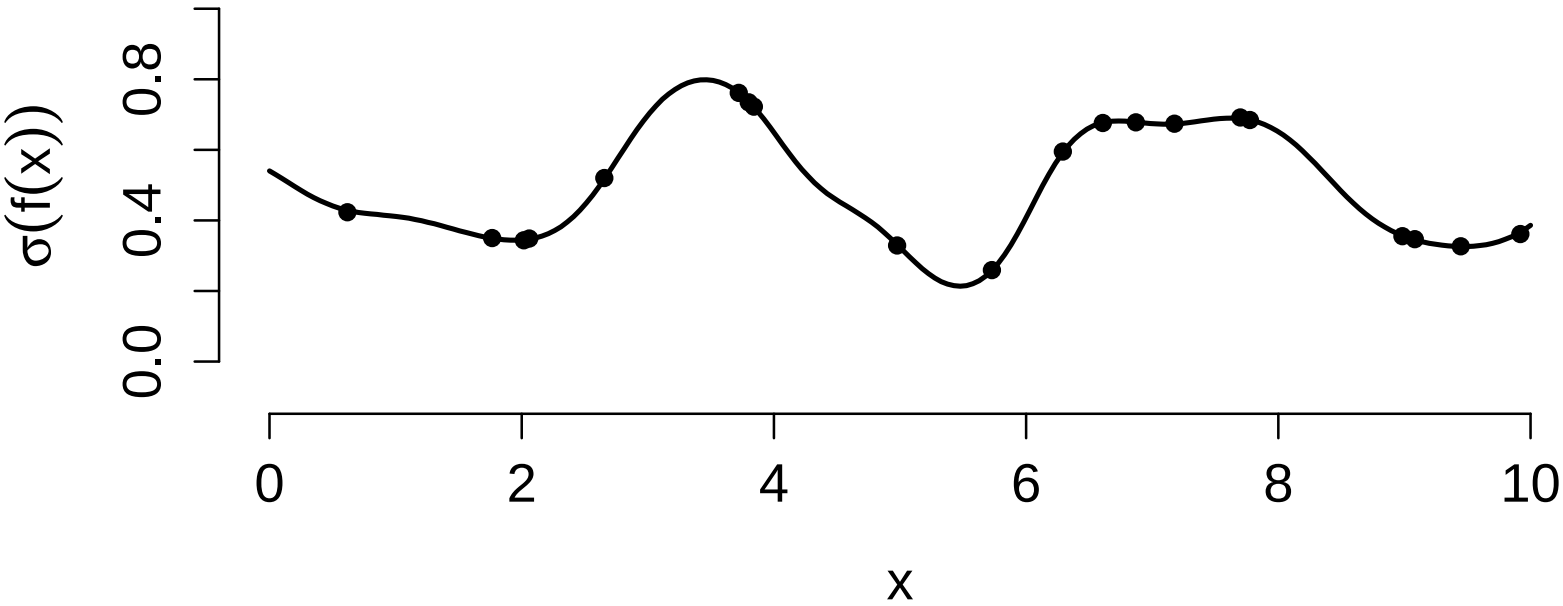


Example: Binary GP classification

$$x_i \overset{iid}{\sim} \text{Unif}(0, 10)$$

$$f(x) \sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ with } k(x_i, x_j) = \exp(-(x_i - x_j)^2/4)$$

$$y_i \overset{ind}{\sim} \text{Bern}\left(\frac{1}{1 + \exp(-f(x_i))}\right)$$

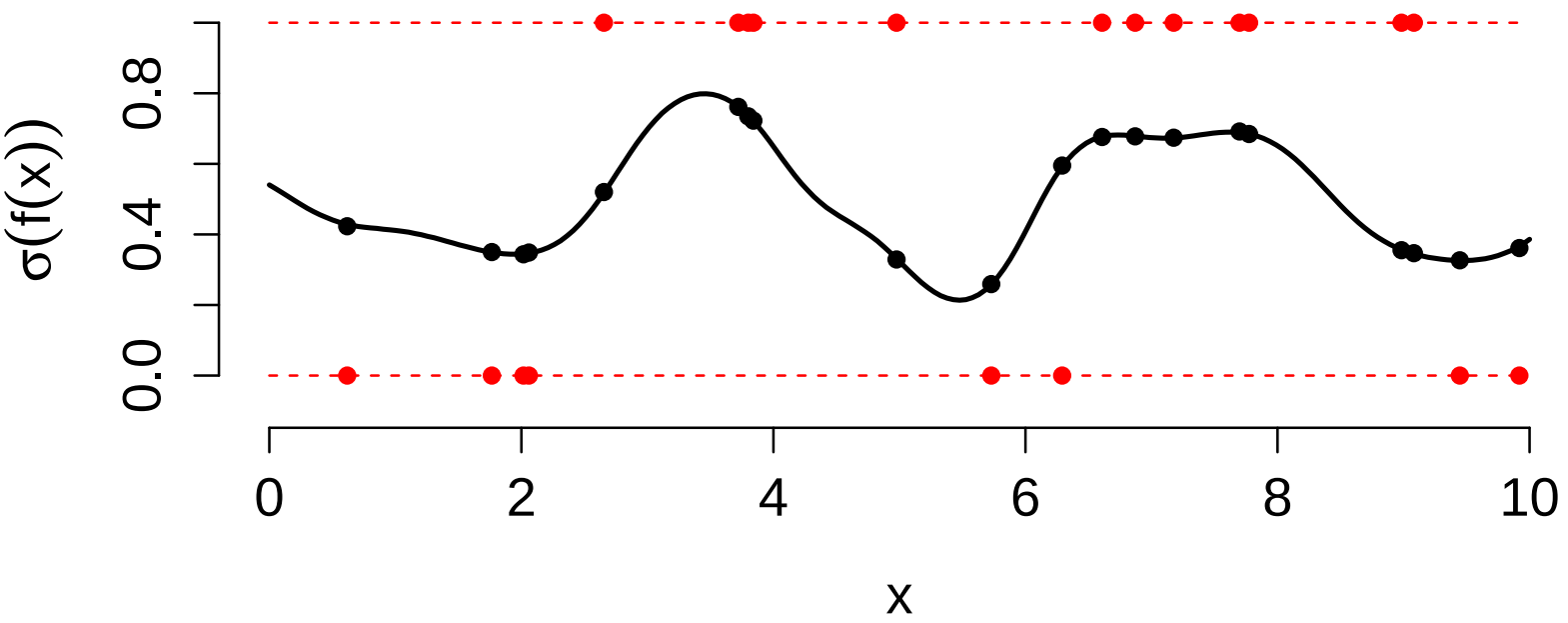


Example: Binary GP classification

$$x_i \overset{iid}{\sim} \text{Unif}(0, 10)$$

$$f(x) \sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ with } k(x_i, x_j) = \exp(-(x_i - x_j)^2/4)$$

$$y_i \overset{ind}{\sim} \text{Bern}\left(\frac{1}{1 + \exp(-f(x_i))}\right)$$

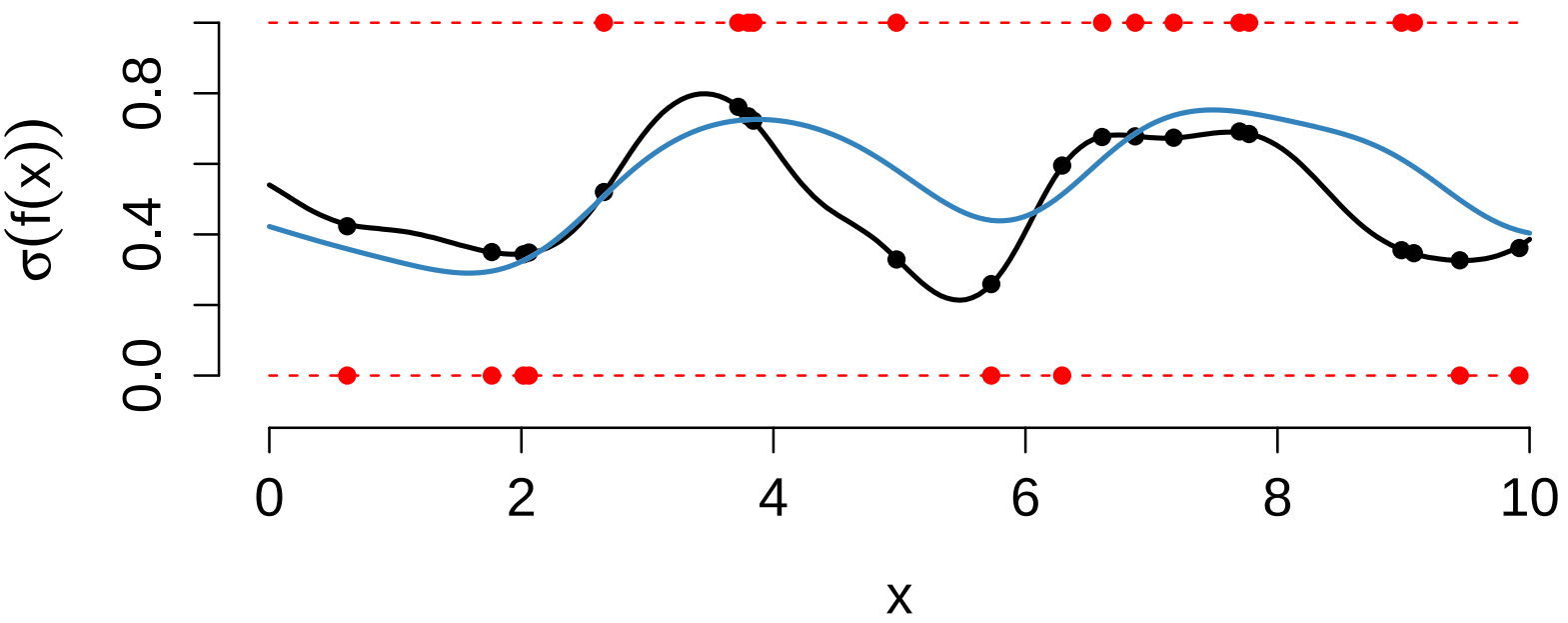


Example: Binary GP classification

$$x_i \overset{iid}{\sim} \text{Unif}(0, 10)$$

$$f(x) \sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ with } k(x_i, x_j) = \exp(-(x_i - x_j)^2/4)$$

$$y_i \overset{ind}{\sim} \text{Bern}\left(\frac{1}{1 + \exp(-f(x_i))}\right)$$

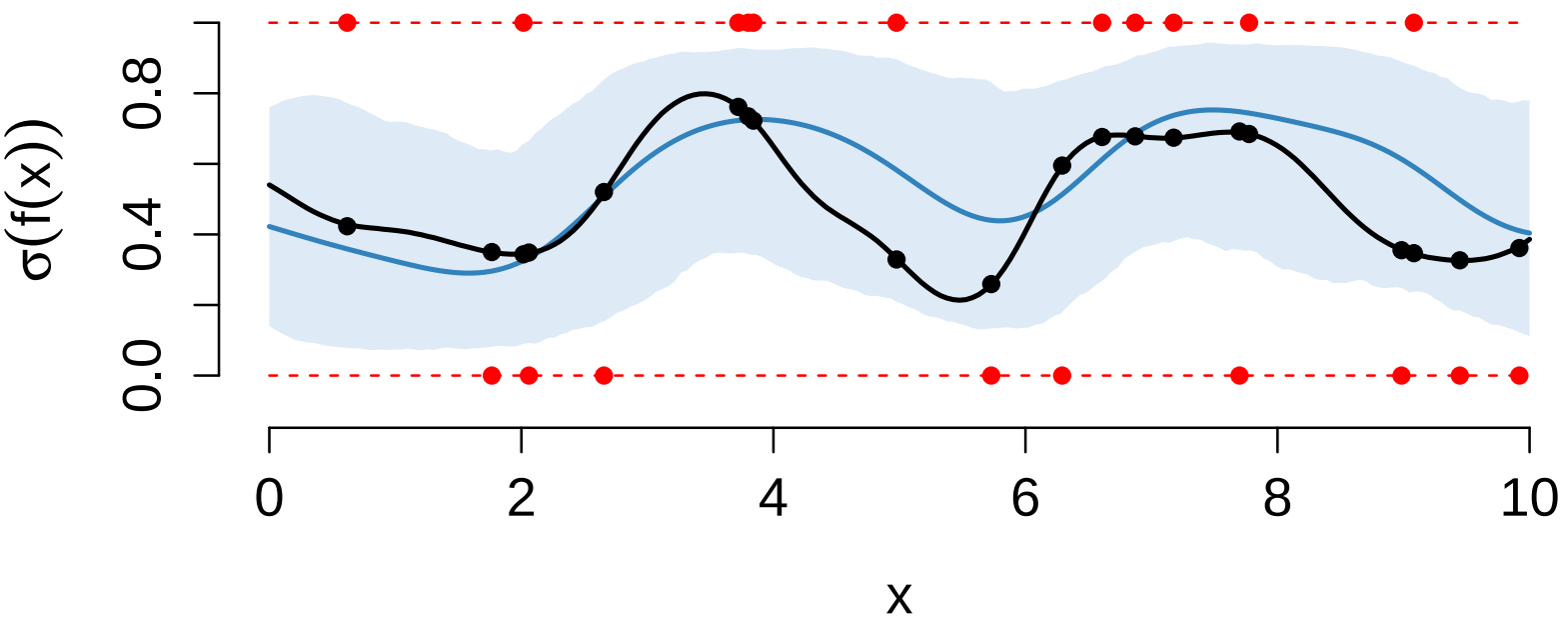


Example: Binary GP classification

$$x_i \overset{iid}{\sim} \text{Unif}(0, 10)$$

$$f(x) \sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ with } k(x_i, x_j) = \exp(-(x_i - x_j)^2/4)$$

$$y_i \overset{ind}{\sim} \text{Bern}\left(\frac{1}{1 + \exp(-f(x_i))}\right)$$



Inference for latent GP models

Two distributions typically of interest

$$\mathbf{f}|\mathbf{y}, \mathbf{X} \propto P(\mathbf{y}|\mathbf{f}, \mathbf{X})P(\mathbf{f}|\mathbf{X}) \qquad \text{(posterior)}$$

$$f^*|\mathbf{y}, \mathbf{X} = \int P(f^*|\mathbf{f}, \mathbf{X}, \mathbf{x}^*)P(\mathbf{f}|\mathbf{y}, \mathbf{X})d\mathbf{f} \qquad \text{(posterior predictive for latent variable)}$$

When $P(y|\mathbf{f})$ is *not* Gaussian, we lack closed-form expression for posterior.

Inference for latent GP models

Two distributions typically of interest

$$\mathbf{f}|\mathbf{y}, \mathbf{X} \propto P(\mathbf{y}|\mathbf{f}, \mathbf{X})P(\mathbf{f}|\mathbf{X}) \quad \text{(posterior)}$$

$$f^*|\mathbf{y}, \mathbf{X} = \int P(f^*|\mathbf{f}, \mathbf{X}, \mathbf{x}^*)P(\mathbf{f}|\mathbf{y}, \mathbf{X})d\mathbf{f} \quad \text{(posterior predictive for latent variable)}$$

When $P(y|\mathbf{f})$ is *not* Gaussian, we lack closed-form expression for posterior. Approaches:

- MCMC (easily extends to parameter inference as well).
- Laplace approximation:
  - Find posterior mode $\hat{\mathbf{f}}$ (using any gradient-based optimizer).
  - Use Normal approximation to posterior

$$P(\mathbf{f}|\mathbf{y}, \mathbf{X}) \approx \mathcal{N}(\hat{\mathbf{f}}, \mathbf{H})$$

    where $\mathbf{H}$ is the negative Hessian of the posterior evaluated at $\hat{\mathbf{f}}$.
- Expectation Propagation, variational approximation.

Applications: climate reconstruction

[M. Tingley and P. Huybers, "Recent temperature extremes at high northern latitudes unprecedented in the past 600 years." *Nature*, 2013]

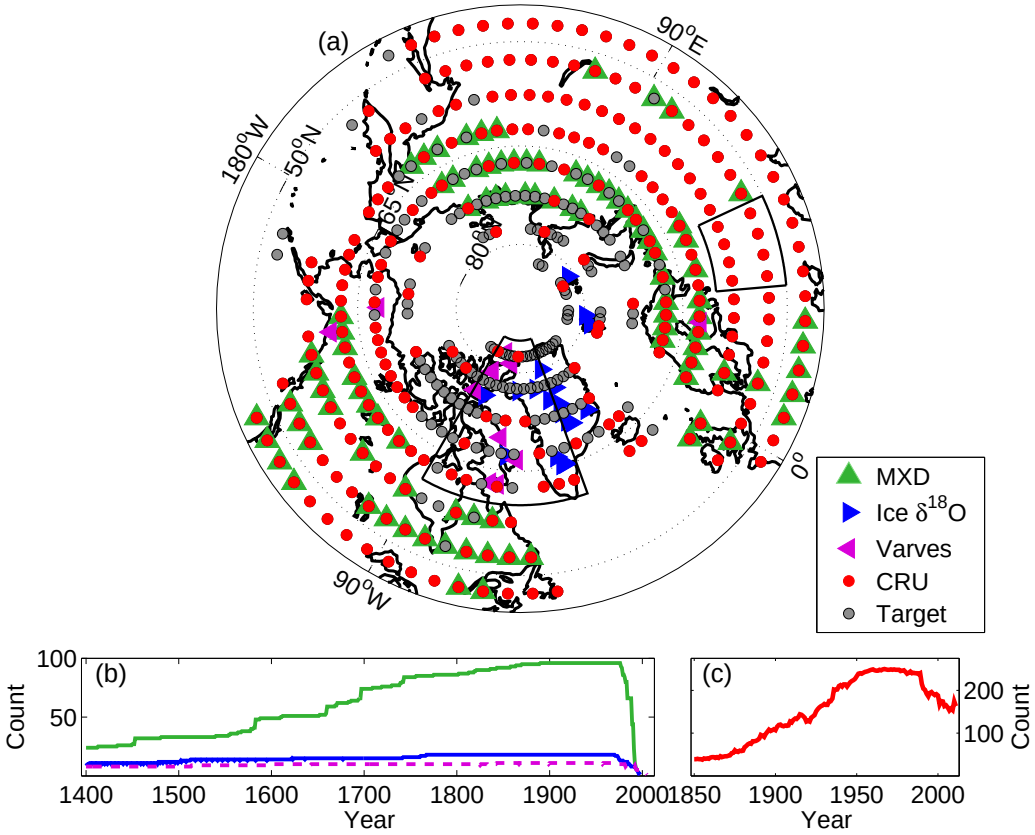


Figure S.1: Data availability in space and time. **(a)** Locations of the data time series. In the legend, MXD refers to the tree ring density series, and Target refers to locations where temperature anomalies are inferred but where there are no observations. The two areas outlined in black are used to assess anomalous warmth in 2010. **(b)** and **(c)** The number and type of proxy (b) and instrumental observations (c) available at each year.

[Tingley & Huybers]

Data

- $\mathbf{O}_i$: temperature data for year i from location set $\mathbf{X}_O$.
- $\mathbf{R}_I$: "proxy" data for year i from location set $\mathbf{X}_R$.

Model

- $\mathbf{T}_i^O$: latent true temperature for year i at locations $\mathbf{X}_O$.
- $\mathbf{O}_i = \mathbf{A}_O \mathbf{T}_i^O$.
- $\mathbf{R}_i = \mathbf{A}_R \mathbf{T}_i^R$.
- $\mathbf{T}_i^R$: latent true temperature for year i at locations $\mathbf{X}_R$.
- $\mathbf{T}_i = (\mathbf{T}_i^O \;\; \mathbf{T}_i^R)$
- $\mathbf{T}_i = \mathbf{\Gamma} \mathbf{T}_{i-1} + \boldsymbol{\eta}_i$.
- $\boldsymbol{\eta} \sim \mathcal{GP}$.

Applications: climate reconstruction

[M. Tingley and P. Huybers, "Recent temperature extremes at high northern latitudes unprecedented in the past 600 years." *Nature*, 2013]

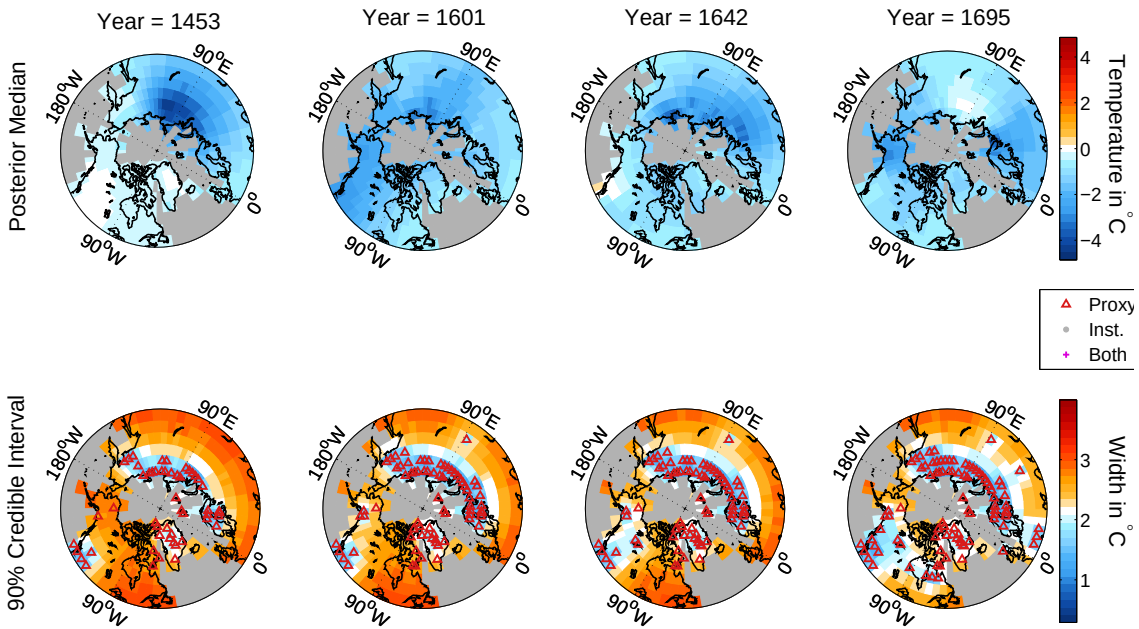


Figure S.8: Temperature anomaly estimates and uncertainties for four years. The top row plots the posterior median of the temperature distribution for each location for 1453, 1601, 1642, and 1695, respectively, while the bottom row plots the widths of the corresponding 90% credible intervals. In the bottom row, symbols denote that a proxy, and/or instrumental observation is available for that location and year.

[Tingley & Huybers]

Data

- $\mathbf{O}_i$: temperature data for year i from location set $\mathbf{X}_O$.
- $\mathbf{R}_I$: "proxy" data for year i from location set $\mathbf{X}_R$.

Model

- $\mathbf{T}_i^O$: latent true temperature for year i at locations $\mathbf{X}_O$.
- $\mathbf{O}_i = \mathbf{A}_O \mathbf{T}_i^O$.
- $\mathbf{R}_i = \mathbf{A}_R \mathbf{T}_i^R$.
- $\mathbf{T}_i^R$: latent true temperature for year i at locations $\mathbf{X}_R$.
- $\mathbf{T}_i = (\mathbf{T}_i^O \;\; \mathbf{T}_i^R)$
- $\mathbf{T}_i = \mathbf{\Gamma} \mathbf{T}_{i-1} + \boldsymbol{\eta}_i$.
- $\boldsymbol{\eta} \sim \mathcal{GP}$.

Applications: species population mapping

[A. Chakraborty et al., "Modeling large scale species abundance with latent spatial processes." *Annals of Applied Statistics*, 2010.]

Data and model:

- $Y(A)$: count of species in region A.
- $Y(A) \sim \text{Pois}(\mu(A))$.
- $\mu(A) = \int_A \lambda(s)ds$.
- $\log(\lambda(s)) = \mathbf{f}(s) + \mathbf{Z}(s)'\beta$.
- $\mathbf{f} \sim \mathcal{GP}$, $\mathbf{Z}$ vector of covariates for location s (e.g. altitude, etc).

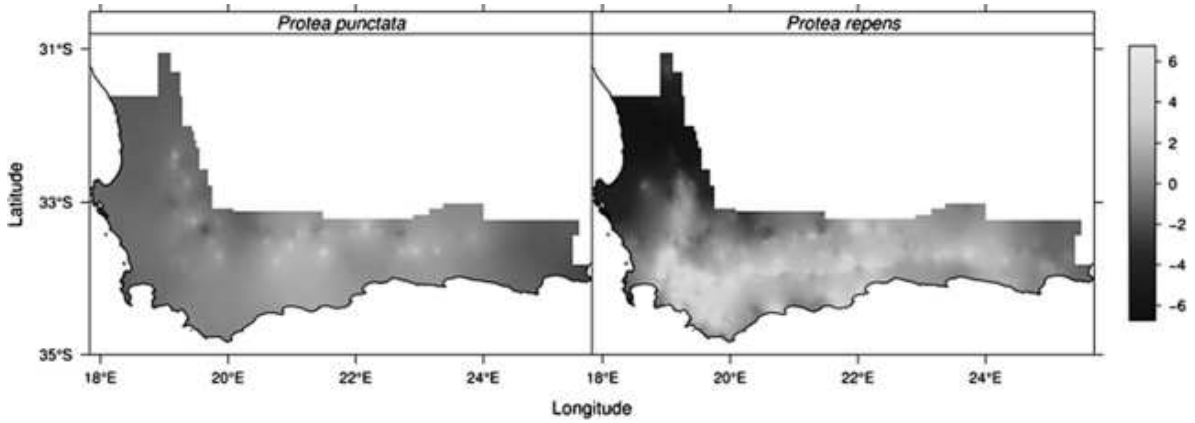


FIG. 6. *Posterior mean spatial effects (θ) for Protea punctata (PRPUNC) and Protea repens (PRREPE). These effects offer local adjustment to potential abundance. Cells with values greater than zero represent regions with larger than expected populations, conditional on the other covariates.*

[Chakraborty et al.]

Applications: computer experiments

[B. Gramacy and H. Lee, "Bayesian treed Gaussian process models with an application to computer modeling." *Journal of the American Statistical Association*, 2008.]

NASA uses computer experiments to simulate the force applied to a vehicle entering the atmosphere: $g(s, \alpha, \beta)$.

- g computed by fluid dynamics simulation; each evaluation takes ~ 20 hours.
- Build GP emulator $f \approx g$ given 3000 observations of g.

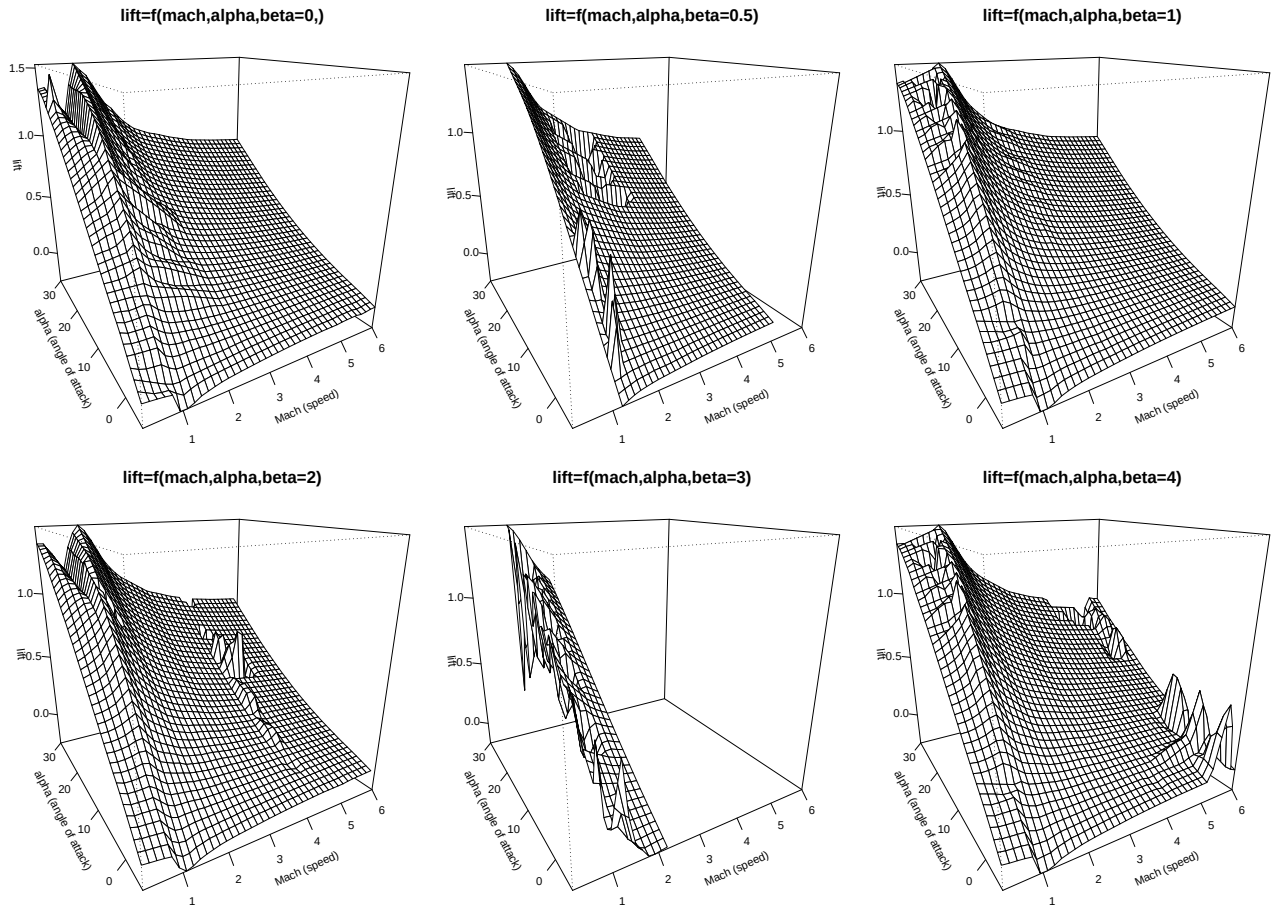


Figure 1: Interpolation of lift by speed and angle of attack for all sideslip levels. Note that for levels 0.5 and 3 (center), Mach ranges only in $(1, 5)$ and $(1.2, 2.2)$.

[Gramacy and Lee]

Applications: Bayesian optimization

[J. Snoek et al. "Practical Bayesian optimization of machine learning algorithms." *NIPS*, 2012.]

Bayesian optimization helps tune hyperparameters for ML algorithms.

- $f(\mathbf{x})$: performance metric (e.g. MSPE) for ML algorithm with tuning parameters $= \mathbf{x}$.
- $f \sim \mathcal{GP}$.

Applications: Bayesian optimization

[J. Snoek et al. "Practical Bayesian optimization of machine learning algorithms." *NIPS*, 2012.]

Bayesian optimization helps tune hyperparameters for ML algorithms.

- $f(\mathbf{x})$: performance metric (e.g. MSPE) for ML algorithm with tuning parameters $= \mathbf{x}$.
- $f \sim \mathcal{GP}$.

Expected improvement in f at $\mathbf{x}^*$:

- Denote $f(\mathbf{x}^*)|\mathbf{f} \sim \mathcal{N}(\mu(\mathbf{x}^*; \mathbf{X}, \mathbf{y}), \sigma(\mathbf{x}^*; \mathbf{X}, \mathbf{y}))$.
- $\gamma(\mathbf{x}^*) = (f(\mathbf{x}_{\text{best}}) - \mu(\mathbf{x}^*; \mathbf{X}, \mathbf{y}))/\sigma(\mathbf{x}^*; \mathbf{X}, \mathbf{y})$, where $\mathbf{x}_{\text{best}} = \text{argmin}_{\mathbf{x}_i} f(\mathbf{x}_i)$.

$$EI(\mathbf{x}^*) = \gamma(\mathbf{x}^*)[1 + \sigma(\mathbf{x}^*; \mathbf{X}, \mathbf{y})\Phi(\gamma(\mathbf{x}^*))]$$

Applications: Bayesian optimization

[J. Snoek et al. "Practical Bayesian optimization of machine learning algorithms." *NIPS*, 2012.]

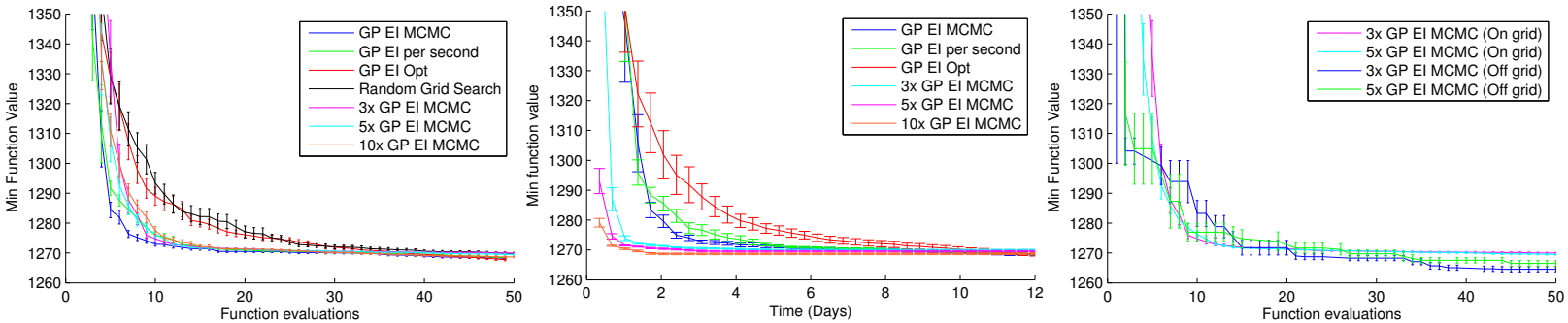
Bayesian optimization helps tune hyperparameters for ML algorithms.

- $f(\mathbf{x})$: performance metric (e.g. MSPE) for ML algorithm with tuning parameters $= \mathbf{x}$.
- $f \sim \mathcal{GP}$.

Expected improvement in f at $\mathbf{x}^*$:

- Denote $f(\mathbf{x}^*)|\mathbf{f} \sim \mathcal{N}(\mu(\mathbf{x}^*; \mathbf{X}, \mathbf{y}), \sigma(\mathbf{x}^*; \mathbf{X}, \mathbf{y}))$.
- $\gamma(\mathbf{x}^*) = (f(\mathbf{x}_{\text{best}}) - \mu(\mathbf{x}^*; \mathbf{X}, \mathbf{y}))/\sigma(\mathbf{x}^*; \mathbf{X}, \mathbf{y})$, where $\mathbf{x}_{\text{best}} = \text{argmin}_{\mathbf{x}_i} f(\mathbf{x}_i)$.

$$EI(\mathbf{x}^*) = \gamma(\mathbf{x}^*)[1 + \sigma(\mathbf{x}^*; \mathbf{X}, \mathbf{y})\Phi(\gamma(\mathbf{x}^*))]$$



(a) (b) (c)

Figure 4: Different strategies of optimization on the Online LDA problem compared in terms of function evaluations (4a), walltime (4b) and constrained to a grid or not (4c).

[Snoek et al.]

Further challenges: non-stationarity

How reasonable is stationarity in practice?

- Boundary effects.
- Mean processes and trends.

Further challenges: non-stationarity

How reasonable is stationarity in practice?

- Boundary effects.
- Mean processes and trends.

Models for non-stationary GPs.

- Basis functions: $\mu(\mathbf{x}) = \sum_{j=1}^{J} w_j m_j(\mathbf{x})$.
- Local approximations and GP trees.
- Dimension expansion: assume in some additional dimensions $\mathbf{z}$, there exists stationary GP $f(\mathbf{x}, \mathbf{z})$.
- Warping: assume there exists g such that $f(g(\mathbf{x}))$ is stationary.

Further challenges: computation

Likelihood calculations and predictions require $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory.

- Simple GP implementations fail on small(ish) ($n > 10000$) data sets!

Further challenges: computation

Likelihood calculations and predictions require $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory.

- Simple GP implementations fail on small(ish) ($n > 10000$) data sets!

Methods for reducing computational complexity:

- Sparsity, including covariance tapering and GMRF approximations to f.
- Structured covariance models (e.g. Kronecker, Toeplitz).
- Low rank covariance models (e.g. inducing points, basis functions).
- Likelihood approximations and approximate inference.