GAUSSIAN PROCESS REGRESSION

CSE 515T

Spring 2015

1. BACKGROUND

The kernel trick again...

The Kernel Trick

Consider again the linear regression model:

$$y(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w} + \varepsilon,$$

with prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \mathbf{\Sigma}).$$

The *kernel trick* is to define the function

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \mathbf{\Sigma} \phi(\mathbf{x}'),$$

which allows us to...

The Kernel Trick

. . . given training data $\mathcal{D}$ and test inputs $\mathbf{X}_*$, write the predictive distribution in this nice form:

$$p(\mathbf{y}_* \mid \mathbf{X}_*, \mathcal{D}, \sigma^2) = \mathcal{N}(\mathbf{y}_*; \boldsymbol{\mu}_{\mathbf{y}_*|\mathcal{D}}, \mathbf{K}_{\mathbf{y}_*|\mathcal{D}}),$$

where

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{y}_*|\mathcal{D}} &= \mathbf{K}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \\ \mathbf{K}_{\mathbf{y}_*|\mathcal{D}} &= \mathbf{K}_{**} - \mathbf{K}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_*,\end{aligned}$$

and we have defined

$$\mathbf{K} = K(\mathbf{X}, \mathbf{X}) \qquad \mathbf{K}_* = K(\mathbf{X}, \mathbf{X}_*) \qquad \mathbf{K}_{**} = K(\mathbf{X}_*, \mathbf{X}_*).$$

The Kernel Trick

This does more than just make the expressions pretty. In particular, it is often *easier/cheaper* to calculate K directly rather than explicitly compute $\phi(\mathbf{x})$ and take the dot product.

Example: "all subsets" kernel.

The Kernel Trick

Idea: completely *abandon* the idea of deriving explicit feature expansions and simply derive (positive-definite) kernel functions K directly!

The Kernel Trick

Maybe we could *skip* the entire procedure of thinking about $\mathbf{w}$, which we never explicitly use here...

$$p(\mathbf{y}_* \mid \mathbf{X}_*, \mathcal{D}, \sigma^2) = \mathcal{N}(\mathbf{y}_*; \mu_{\mathbf{y}_*|\mathcal{D}}, K_{\mathbf{y}_*|\mathcal{D}}),$$

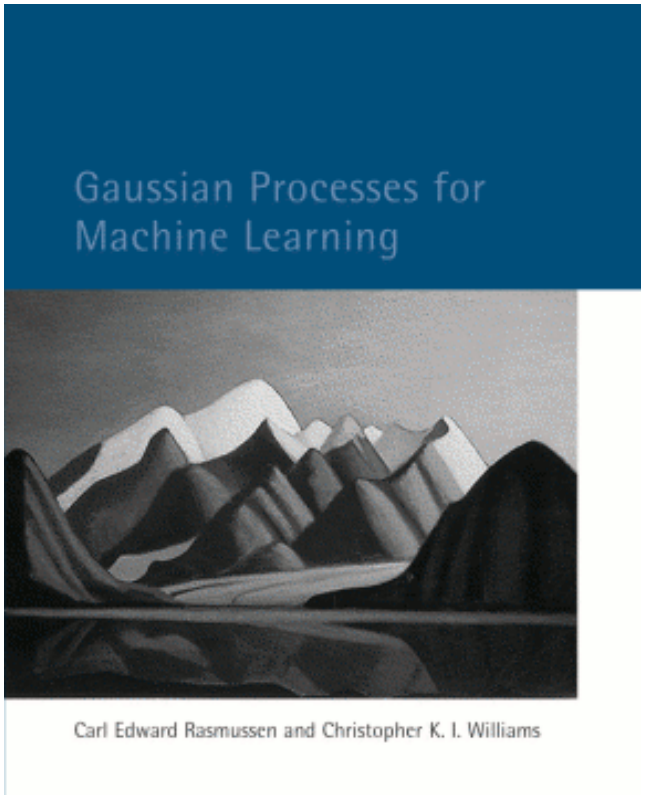
where

$$\mu_{\mathbf{y}_*|\mathcal{D}} = \mathbf{K}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$
$$K_{\mathbf{y}_*|\mathcal{D}} = \mathbf{K}_{**} - \mathbf{K}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_*.$$

2. GAUSSIAN PROCESSES

A reimagining of Bayesian regression

For more information



`http://www.gaussianprocess.org/`

(Also code!)

Regression

Consider the general *regression* problem. Here we have:

- an input domain $\mathcal{X}$ (for example, $\mathbb{R}^n$, but in general anything),
- an unknown function $f\colon \mathcal{X} \to \mathbb{R}$, and
- and (perhaps noisy) observations of the function: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, where $y_i = f(\mathbf{x}_i) + \varepsilon_i$.

Our goal is to *predict* the value of the function $f(\mathbf{X}_*)$ at some test locations $\mathbf{X}_*$.

Gaussian processes

- Gaussian processes take a *nonparameteric* approach to regression. We select a *prior distribution* over the function f and condition this distribution on our observations, using the posterior distribution to make predictions.
- Gaussian processes are very *powerful* and leverage the many *convenient properties* of the Gaussian distribution to enable tractable inference.

From the Gaussian distribution to GPs

- How can we leverage these useful properties of the Gaussian distribution to approach the regression problem? We have a problem: the latent function f is usually *infinite dimensional*; however, the multivariate Gaussian distribution is only useful in *finite dimensions.*
- The Gaussian process is a *natural generalization* of the multivariate Gaussian distribution to *potentially infinite settings.*

GPs: Definition

Definition (GPs)

A Gaussian process is a (potentially infinite) collection of random variables such that the joint distribution of any finite number of them is multivariate Gaussian.

GPs: Notation

A Gaussian process distribution on f is written

$$p(f) = \mathcal{GP}(f; \mu, K),$$

and just like the multivariate Gaussian distribution, is parameterized by its first two moments (now functions):

- $\mathbb{E}[f] = \mu \colon \mathcal{X} \to \mathbb{R}$, the *mean function,* and
- $\mathbb{E}\Big[\big(f(x) - \mu(x)\big)\big(f(x') - \mu(x')\big)\Big] = K \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, a positive semidefinite *covariance function* or *kernel.*

GPs: Mean and covariance functions

- The mean function encodes the *central tendency* of the function, and is often assumed to be a constant (usually zero).
- The covariance function encodes information about the *shape* and *structure* we expect the function to have. A simple and very common example is the *squared exponential* covariance:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-1/2\|\mathbf{x} - \mathbf{x}'\|^2\right),$$

which encodes the notation that "nearby points should have similar function values."

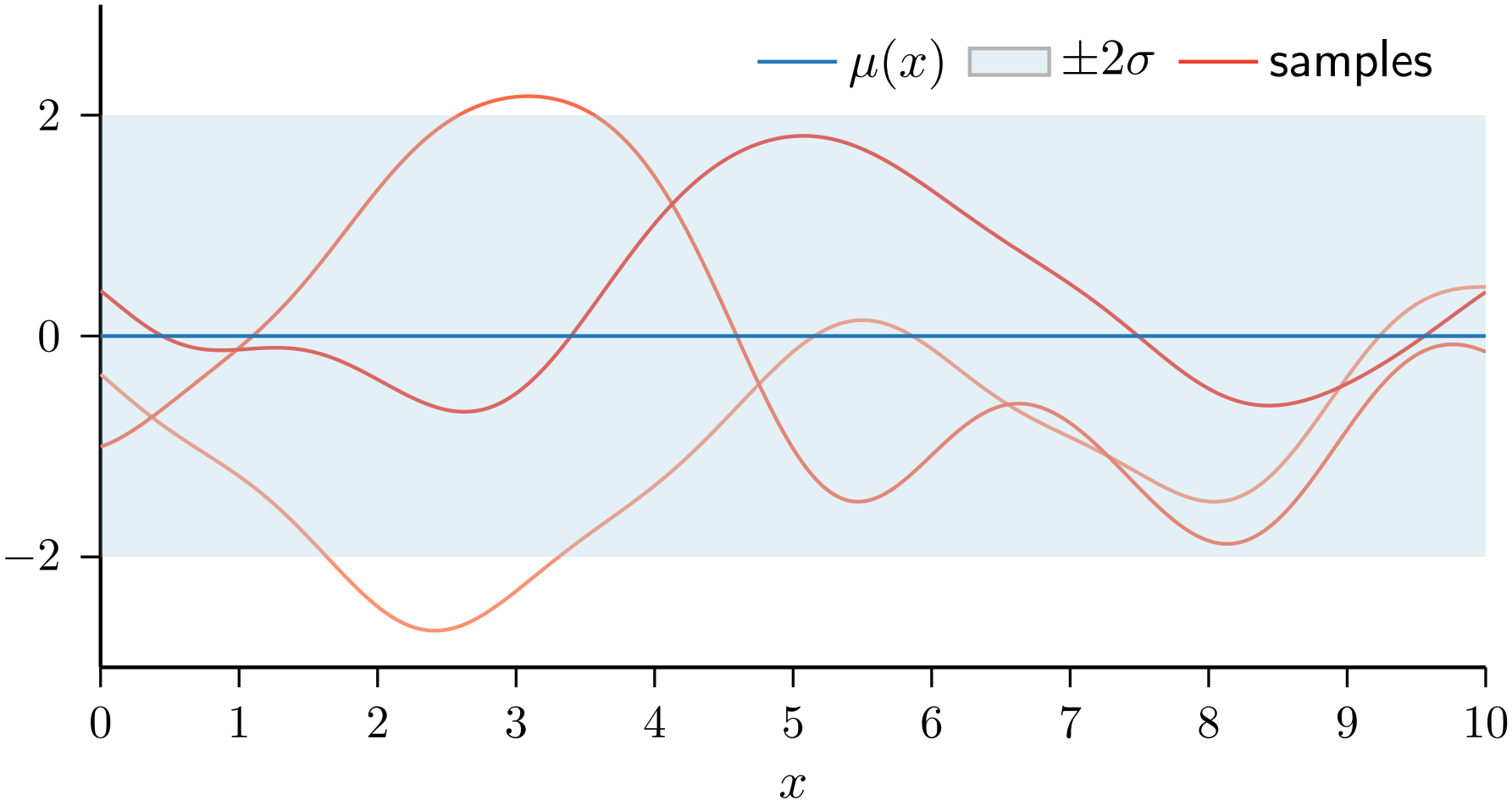
GPs: Prior on finite sets

Suppose we have selected a GP prior $\mathcal{GP}(f; \mu, K)$ for the function f. Consider a finite set of points $\mathbf{X} \subseteq \mathcal{X}$. The GP prior on f, by definition, *implies* the following joint distribution on the associated function values $\mathbf{f} = f(\mathbf{X})$:

$$p(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}\big(\mathbf{f}; \mu(\mathbf{X}), K(\mathbf{X}, \mathbf{X})\big).$$

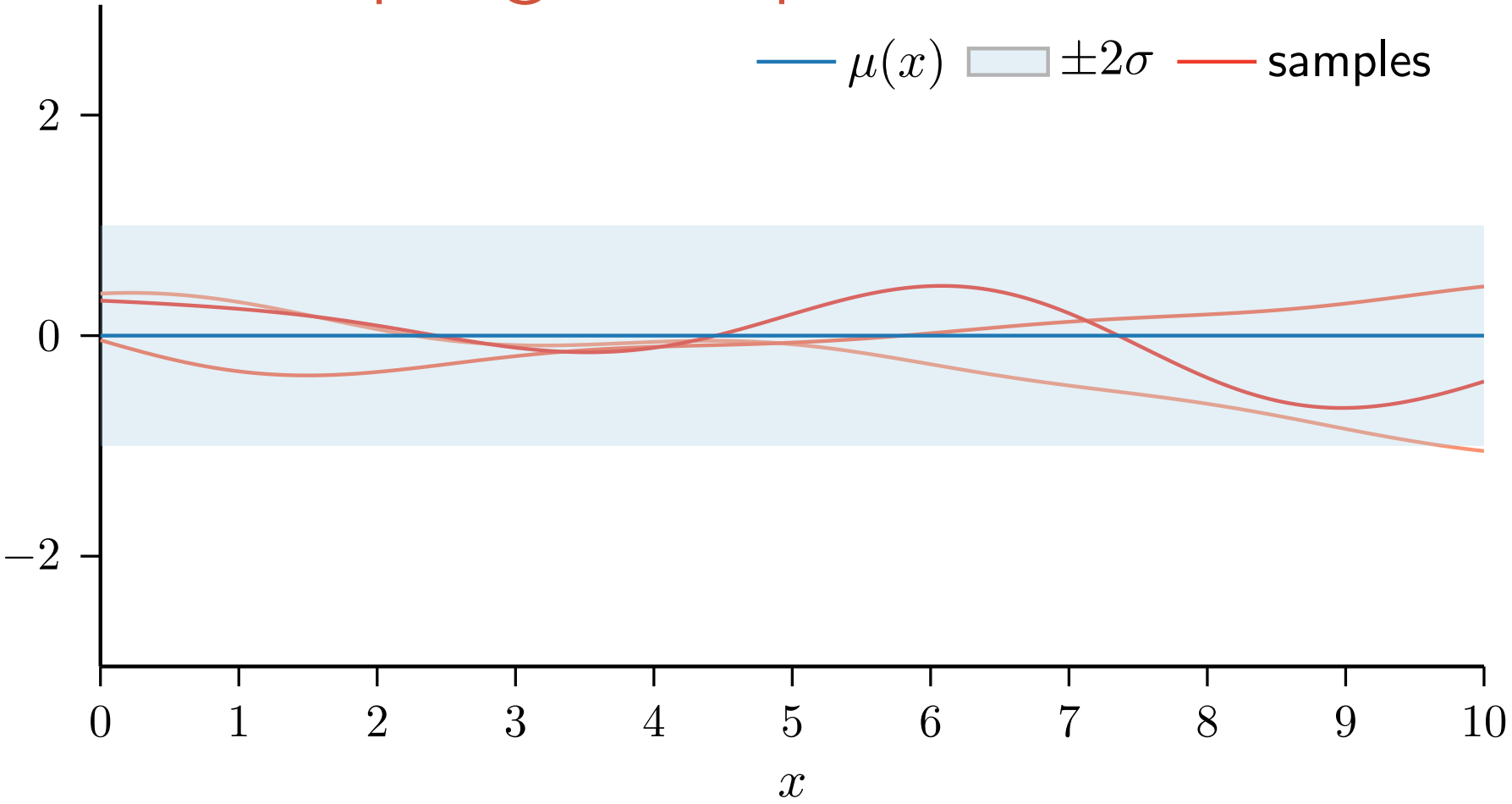
That is, we simply evaluate the mean and covariance functions at $\mathbf{X}$ and take the associated multivariate Gaussian distribution. Very simple!

Prior: Sampling examples



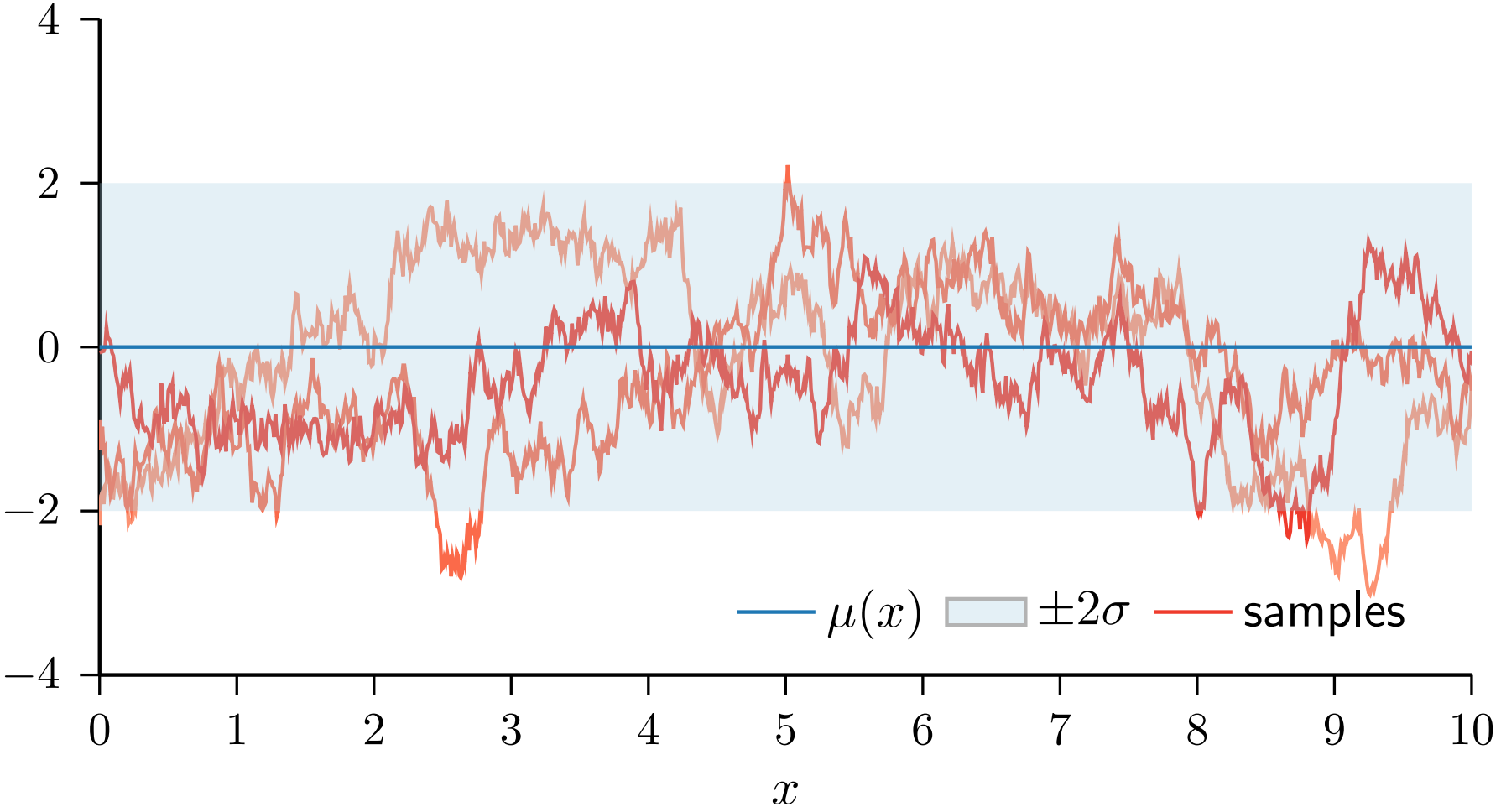
$$K = \exp\left(-1/2\|x - x'\|^2\right)$$

Prior: Sampling examples



$$K = \lambda^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right) \qquad \lambda = 1/2, \ell = 2$$

Prior: Sampling examples



$$K = \exp\left(-\|x - x'\|\right)$$

From the prior to the posterior

So far, I've only told you how to construct *prior* distributions over the function f. How do we *condition* our prior on some observations $\mathcal{D} = (\mathbf{X}, \mathbf{f})$ to *make predictions* about the value of f at some points $\mathbf{X}_*$?

From the prior to the posterior

We begin by writing the *joint distribution* between the training function values $f(\mathbf{X}) = \mathbf{f}$ and the test function values $f(\mathbf{X}_*) = \mathbf{f}_*$:

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix}; \begin{bmatrix} \mu(\mathbf{X}) \\ \mu(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right) \cdots$$

From the prior to the posterior

. . . we then *condition* this multivariate Gaussian on the known training values $\mathbf{f}$. We already know how to do that!

$$p(\mathbf{f}_* \mid \mathbf{X}_*, \mathcal{D}) = \mathcal{N}\big(\mathbf{f}_*; \mu_{f|\mathcal{D}}(\mathbf{X}_*), K_{f|\mathcal{D}}(\mathbf{X}_*, \mathbf{X}_*)\big),$$

where

$$\mu_{f|\mathcal{D}}(\mathbf{x}) = \mu(\mathbf{x}) + K(\mathbf{x}, \mathbf{X})\mathbf{K}^{-1}\big(\mathbf{f} - \mu(\mathbf{X})\big)$$
$$K_{f|\mathcal{D}}(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X})\mathbf{K}^{-1}K(\mathbf{X}, \mathbf{x}').$$

From the prior to the posterior

Notice that the functions $\mu_{f|\mathcal{D}}$ and $K_{f|\mathcal{D}}$ are *valid mean and covariance* functions, respectively. That means the previous slide is telling us the posterior distribution over f is a Gaussian process!

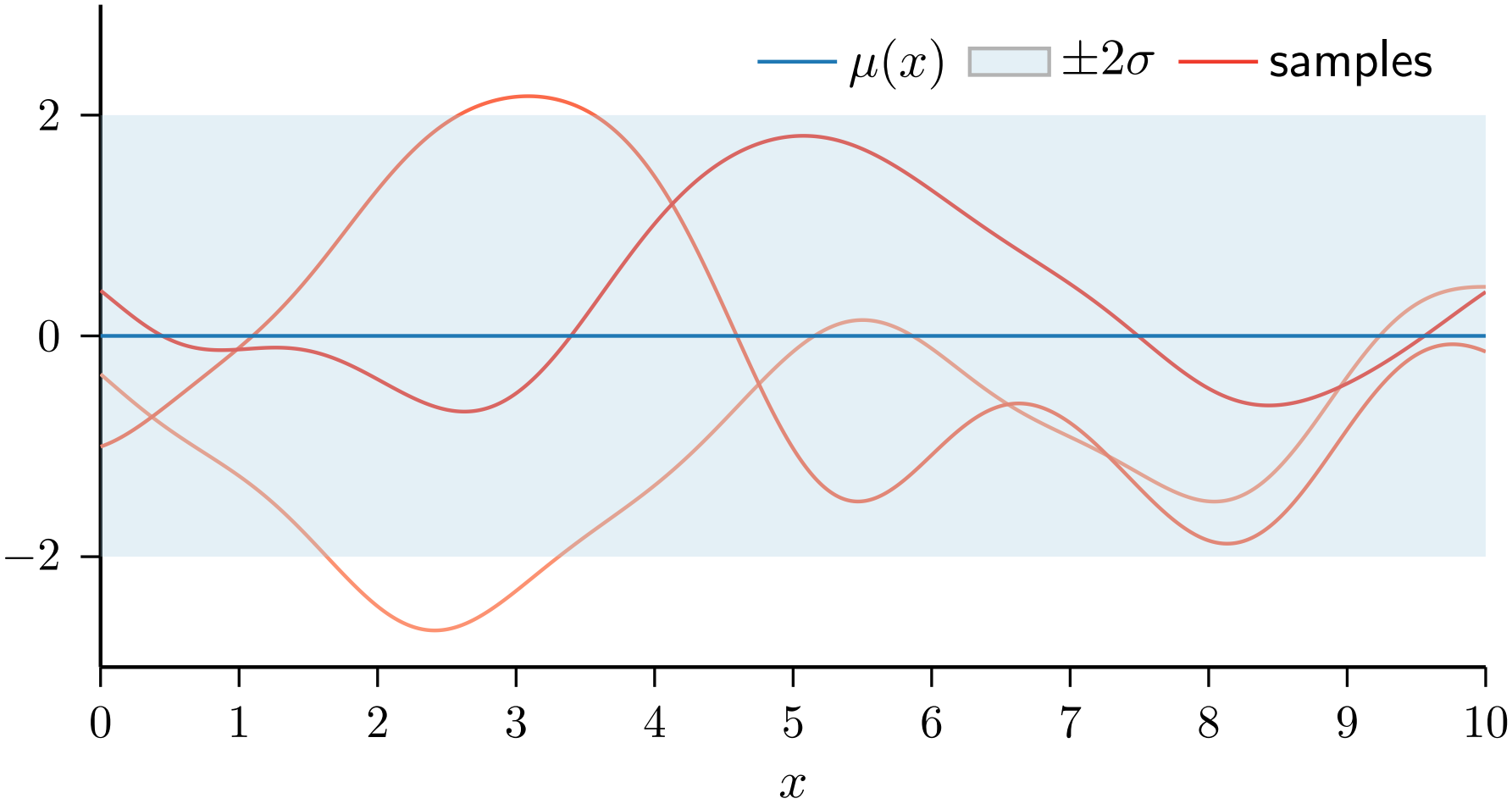
The posterior mean

One way to understand the posterior mean function $\mu_{f|\mathcal{D}}$ is as a *correction to the prior mean* consisting of a *weighted combination* of kernel functions, one for each training data point:

$$\begin{aligned}\mu_{f|D}(\mathbf{x}) &= \mu(\mathbf{x}) + K(\mathbf{x}, \mathbf{X})\big(K(\mathbf{X}, \mathbf{X})\big)^{-1}\big(\mathbf{f} - \mu(\mathbf{X})\big) \\ &= \mu(\mathbf{x}) + \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}),\end{aligned}$$

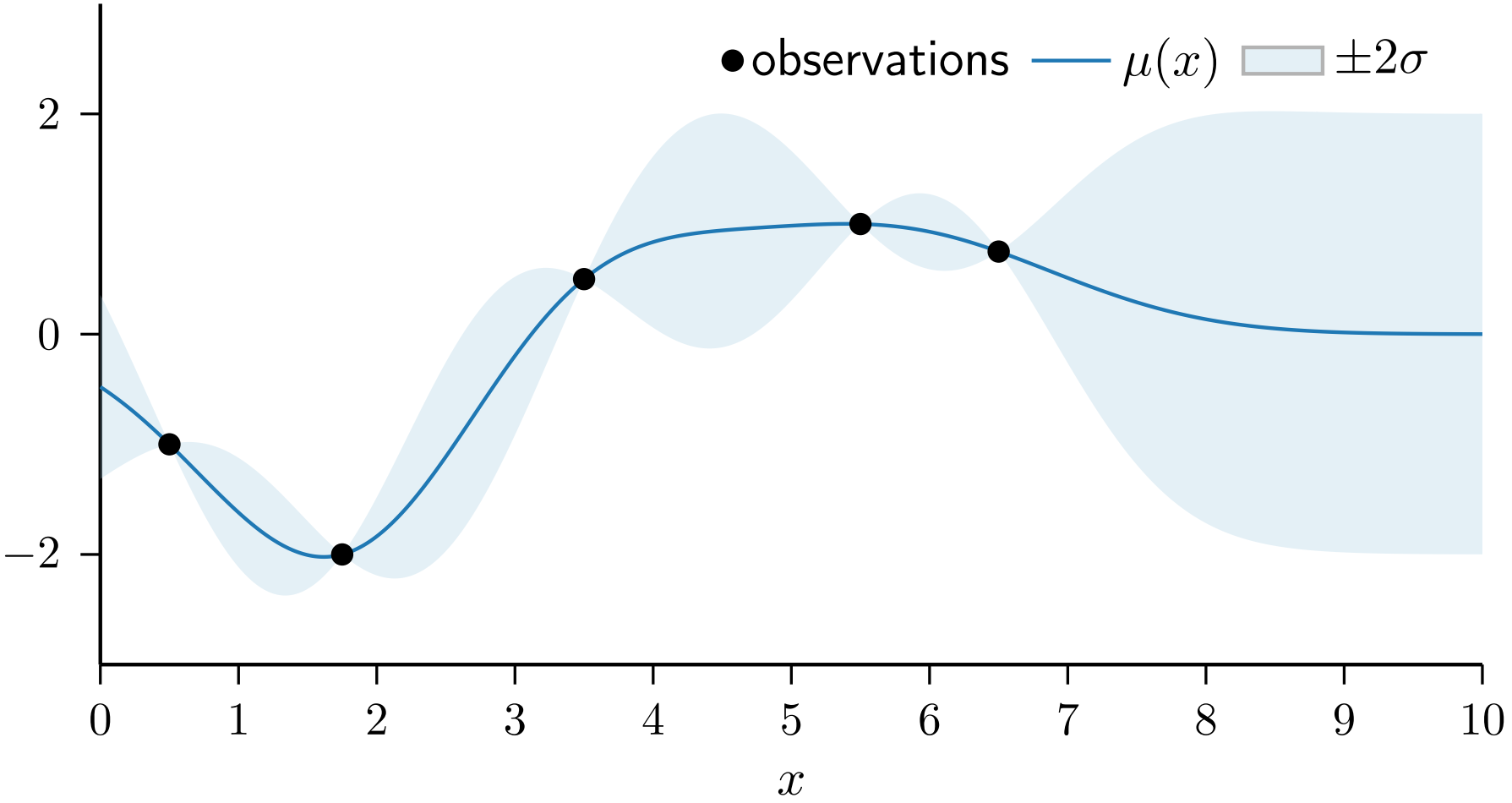
where $\alpha_i = K(\mathbf{X}, \mathbf{X})^{-1}\big(f(\mathbf{x}_i) - \mu(\mathbf{x}_i)\big)$.

Prior

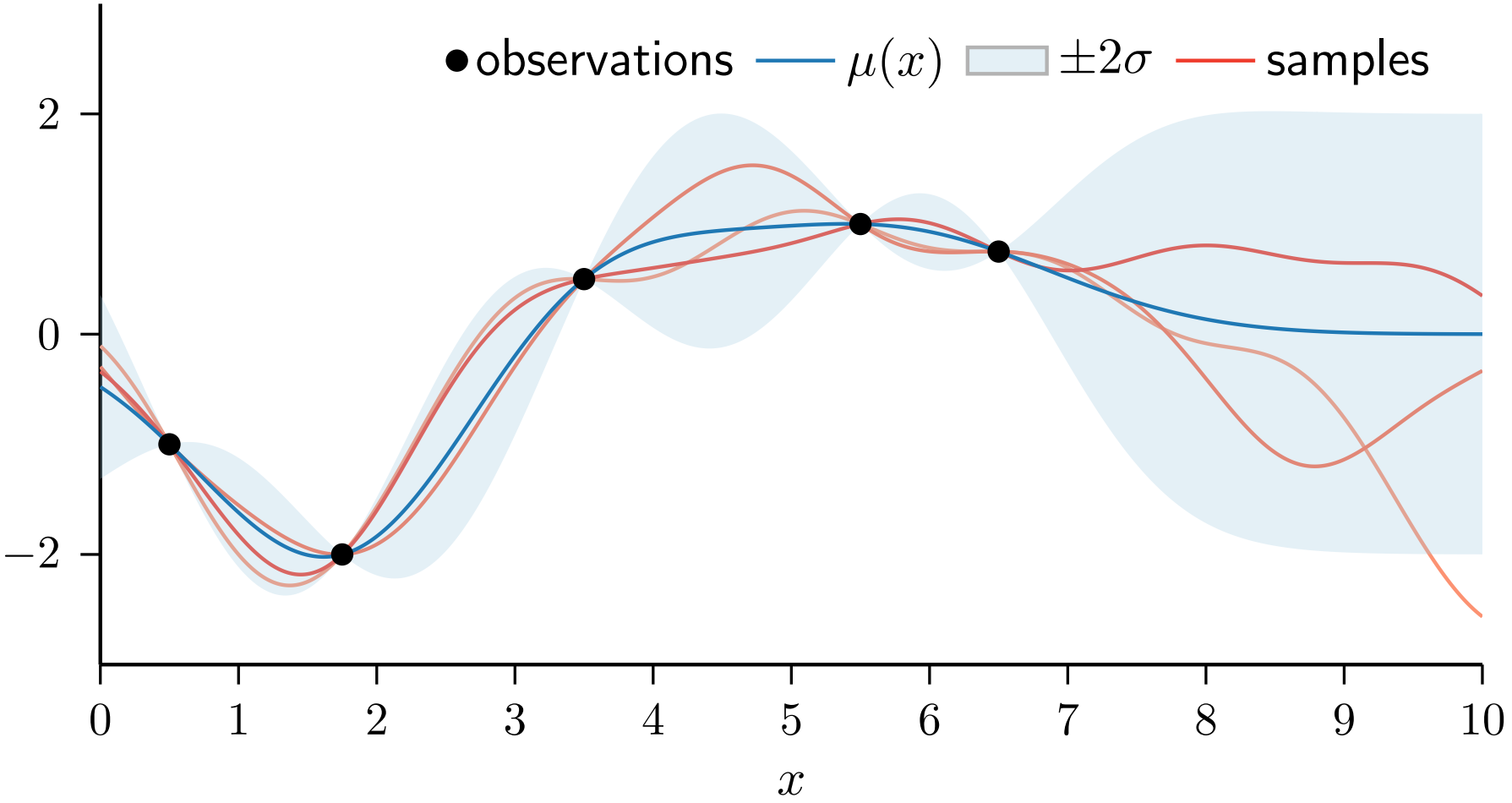


$$K = \exp\left(-1/2\|x - x'\|^2\right)$$

Posterior example



Posterior: Sampling



Dealing with noise

So far, we have assumed we can sample the function f *exactly*, which is uncommon in regression settings. How do we deal with *observation noise?*

tl;dr: the same way we did with Bayesian linear regression!

Dealing with noise

We must create a *model* for our observations given the latent function. To begin, we will choose the simple iid, zero-mean additive Gaussian noise model:

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon,$$
$$p(\varepsilon \mid \mathbf{x}) = \mathcal{N}(\varepsilon; 0, \sigma^2);$$

combined we have

$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{y}; \mathbf{f}, \sigma^2 \mathbf{I}).$$

Noisy posterior

To derive the posterior given *noisy observations* $\mathcal{D}$, we again write the joint distribution between the training function values $\mathbf{y}$ and the test function values $\mathbf{f}_*$:

$$p(\mathbf{y}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix}; \begin{bmatrix} \mu(\mathbf{X}) \\ \mu(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right) \ldots \tag{1}$$

Noisy posterior

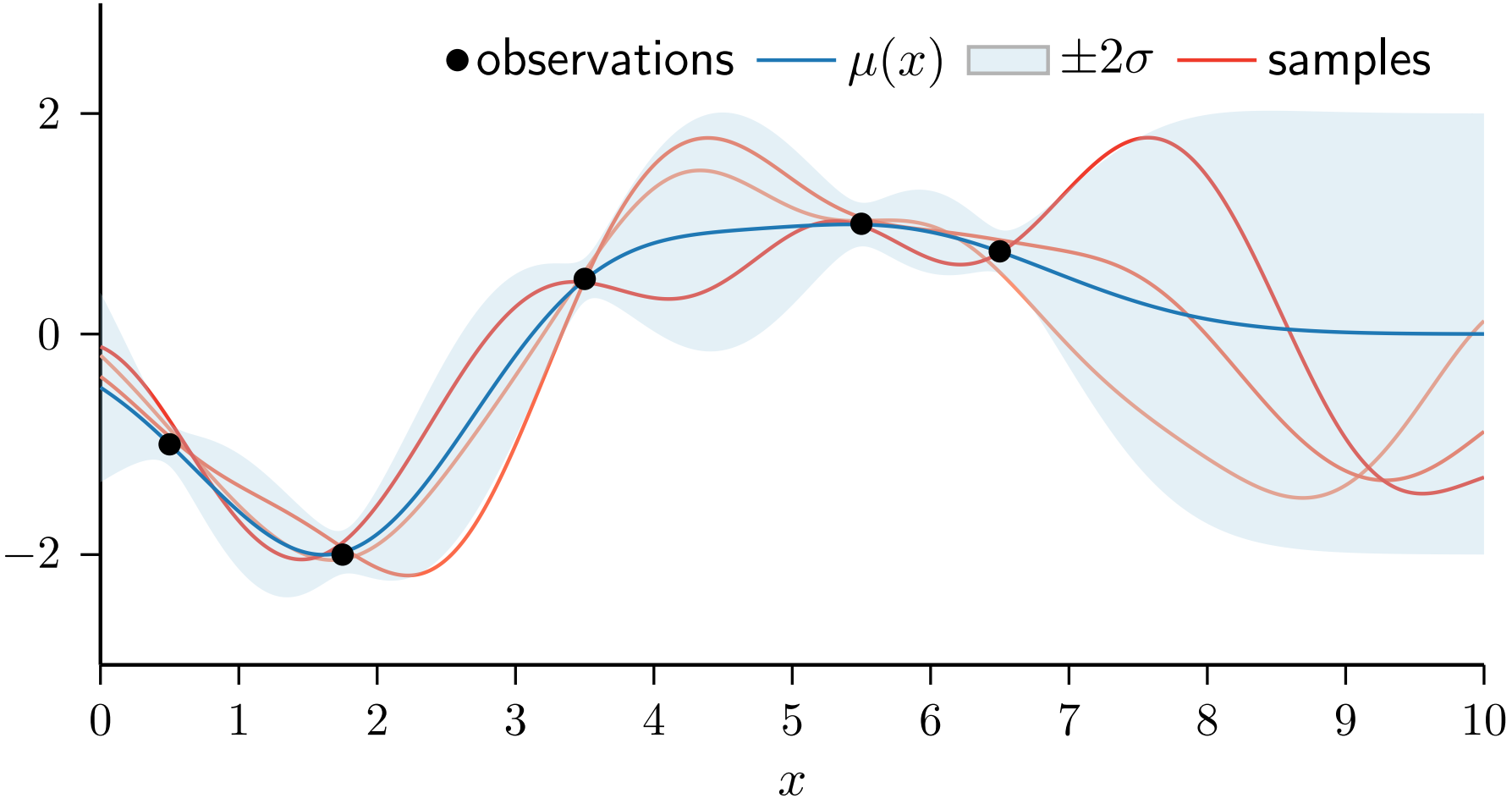
. . . and *condition* as before.

$$p(\mathbf{f}_* \mid \mathbf{X}_*, \mathcal{D}) = \mathcal{N}\big(\mathbf{f}_*; \mu_{f|\mathcal{D}}(\mathbf{X}_*), K_{f|\mathcal{D}}(\mathbf{X}_*, \mathbf{X}_*)\big),$$

where

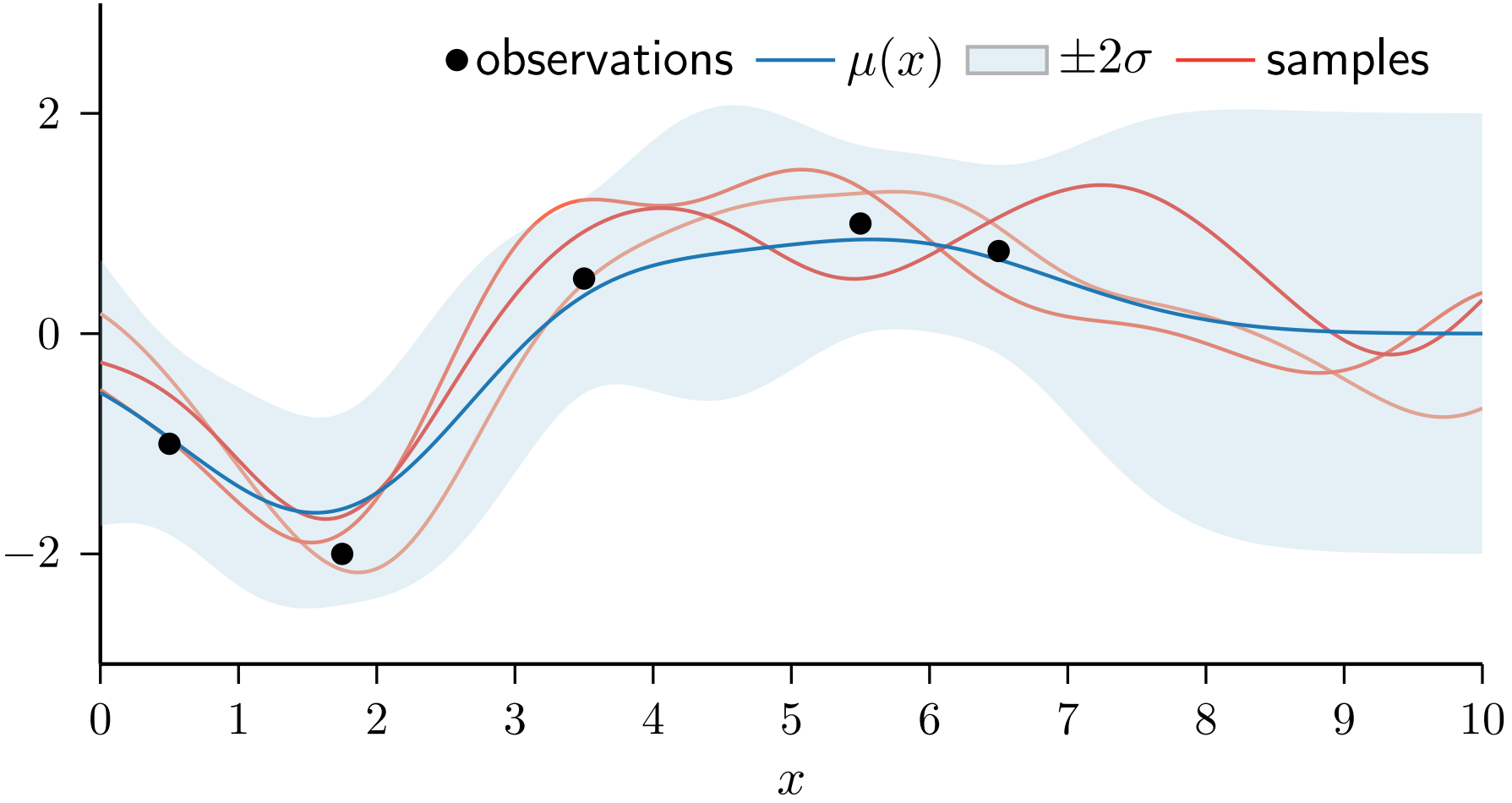
$$\mu_{f|\mathcal{D}}(\mathbf{x}) = \mu(\mathbf{x}) + K(\mathbf{x}, \mathbf{X})\big(K(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}\big)^{-1}\big(\mathbf{y} - \mu(\mathbf{X})\big)$$
$$K_{f|\mathcal{D}}(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X})\big(K(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}\big)^{-1}K(\mathbf{X}, \mathbf{x}').$$

Noisy posterior: Sampling



$\sigma = 0.1$

Noisy posterior: Sampling



$$\sigma = 0.5$$

3. HYPERPARAMETERS

We're not done yet. . .

Hyperparameters

- So far, we have assumed that the Gaussian process prior distribution on f has been specified *a priori.*
- But this prior distribution *itself* has parameters, for example the length scale ℓ, the output scale λ, and the noise variance σ^2. As parameters of a prior distribution, we call these *hyperparameters.*
- For convenience, we will write θ to denote the vector of all hyperparameters of the model (including of μ and K).
- How do we *learn* θ?

Marginal likelihood

Assume we have chosen a parameterized prior

$$p(f \mid \theta) = \mathcal{GP}\big(f; \mu(\mathbf{x}; \theta), K(\mathbf{x}, \mathbf{x}'; \theta)\big).$$

We will measure the quality of the fit to our training data $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ with the *marginal likelihood,* the probability of *observing the given data* under our prior:

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

where we have *marginalized* the unknown function values $\mathbf{f}$ (hence, marginal likelihood).

Marginal likelihood: Evaluating

Thankfully, this is an integral we can do *analytically* under the Gaussian noise assumption!

$$
\begin{aligned}
p(\mathbf{y} \mid \mathbf{X}, \theta) &= \int p(\mathbf{y} \mid \mathbf{f})\, p(\mathbf{f} \mid \mathbf{X}, \theta)\, \mathrm{d}\mathbf{f}, \\
&= \int \overbrace{\mathcal{N}(\mathbf{y}; \mathbf{f}, \sigma^2 \mathbf{I})}^{\text{iid noise}} \overbrace{\mathcal{N}\big(\mathbf{f}; \mu(\mathbf{X}; \theta), K(\mathbf{X}, \mathbf{X}; \theta)\big)}^{\text{GP prior}}\, \mathrm{d}\mathbf{f} \\
&= \mathcal{N}\big(\mathbf{y}; \mu(\mathbf{X}; \theta), K(\mathbf{X}, \mathbf{X}; \theta) + \sigma^2 \mathbf{I}\big).
\end{aligned}
$$

(Convolutions of two Gaussians are Gaussian.)

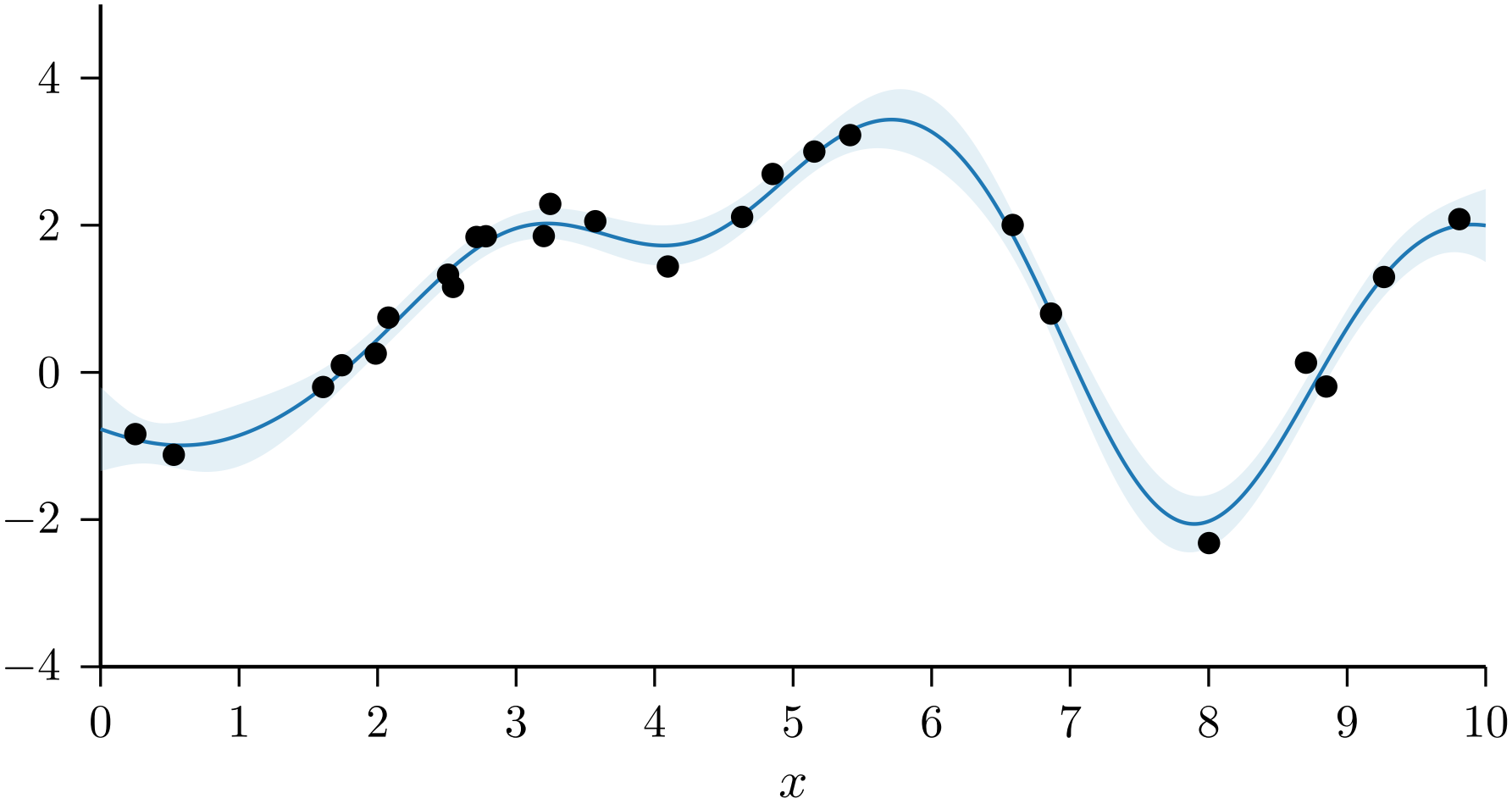
Marginal likelihood: Evaluating

The log-likelihood of our data under the chosen prior are then (writing $\mathbf{V} = (K(\mathbf{X}, \mathbf{X}; \theta) + \sigma^2 \mathbf{I})$):

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = \underbrace{-\frac{(\mathbf{y} - \boldsymbol{\mu})^\top \mathbf{V}^{-1} (\mathbf{y} - \boldsymbol{\mu})}{2}}_{\text{data fit}} \underbrace{- \frac{\log \det \mathbf{V}}{2}}_{\text{Occam's razor}} - \frac{N \log 2\pi}{2}$$

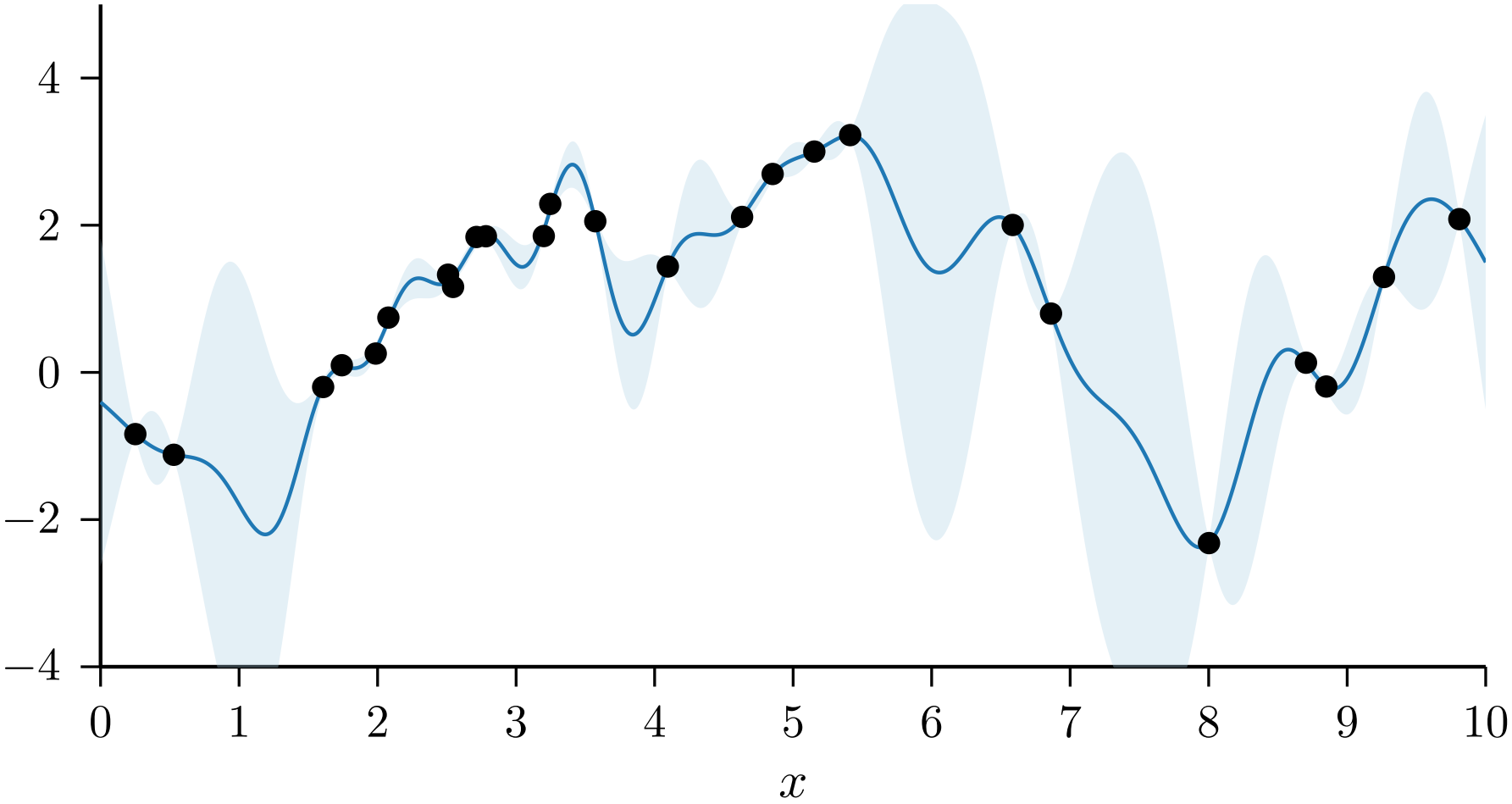
The first term is large when the *data fit the model well*, and the second term is large when the *volume of the prior covariance is small;* that is, when the model is *simpler*.

Hyperparameters: Example



$$\theta = (\lambda, \ell, \sigma) = (1, 1, 1/5), \quad \log p(\mathbf{y} \mid \mathbf{X}, \theta) = -27.6$$

Hyperparameters: Example



$$\theta = (\lambda, \ell, \sigma) = (2, 1/3, 1/20), \quad \log p(\mathbf{y} \mid \mathbf{X}, \theta) = -46.5$$

Hyperparameters are important

Comparing the marginal likelihoods, we see that the observed data are *over 100 million times more likely* to have been generated by the first model rather than from the second model! Clearly hyperparameters can be *quite important.*

Can we marginalize hyperparameters?

To be fully Bayesian, we would choose a *hyperprior* over θ, $p(\theta)$, and *marginalize* the unknown hyperparameters when making predictions:

$$p(f_* \mid \mathbf{x}_*, \mathcal{D}) = \frac{\int p(f_* \mid \mathbf{x}_*, \mathcal{D}, \theta) p(\mathbf{y} \mid \mathbf{X}, \theta) p(\theta) \, \mathrm{d}\theta}{\int p(\mathbf{y} \mid \mathbf{X}, \theta) p(\theta) \, \mathrm{d}\theta}$$

Unfortunately, this integral *cannot* be resolved analytically.

(Of course. . .)

Maximum likelihood-II

- Instead, if we believe the posterior distribution over θ to be *well-concentrated* (for example, if we have many training examples), we may approximate $p(\theta \mid \mathcal{D})$ with a *delta* distribution at the point with the *maximum marginal likelihood:*

$$\theta_{\mathsf{MLE}} = \arg\max_{\theta} p(\mathbf{y} \mid \mathbf{X}, \theta).$$

  This is called *maximum likelihood-II* (ML-II) inference.
- This effectively gives the approximation

$$p(f_* \mid \mathbf{x}_*, \mathcal{D}) \approx p(f_* \mid \mathbf{x}_*, \mathcal{D}, \theta_{\mathsf{MLE}}).$$

- How can we find θ_{MLE}?