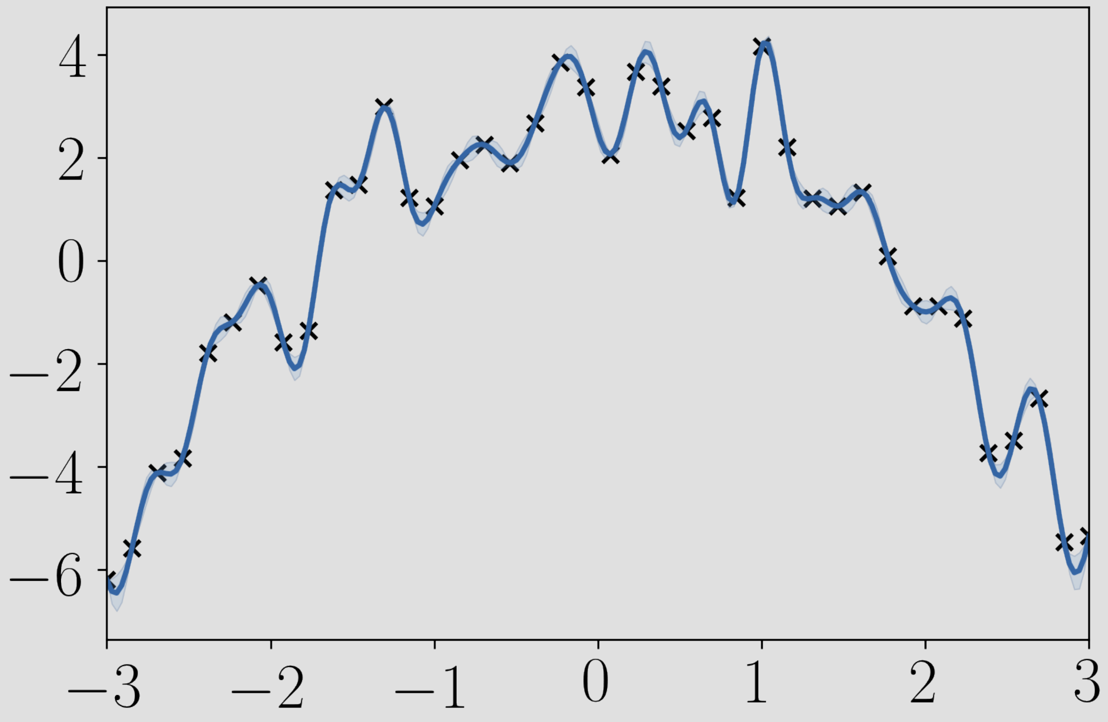
Nuances of GP

Noisy observations



Noisy observations

$$\widehat{f}(x) = f(x) + \epsilon$$

Independent Gaussian noise

$$\epsilon \sim \mathcal{N}(0, s^2)$$

Noisy observations

$$\widehat{f}(x) = f(x) + \epsilon$$ Independent Gaussian noise

$$\epsilon \sim \mathcal{N}(0, s^2)$$

$$m(x) = 0$$

$$\widehat{K}(x_i - x_j) = K(x_i - x_j) + s^2 \mathbb{I}[x_i = x_j]$$

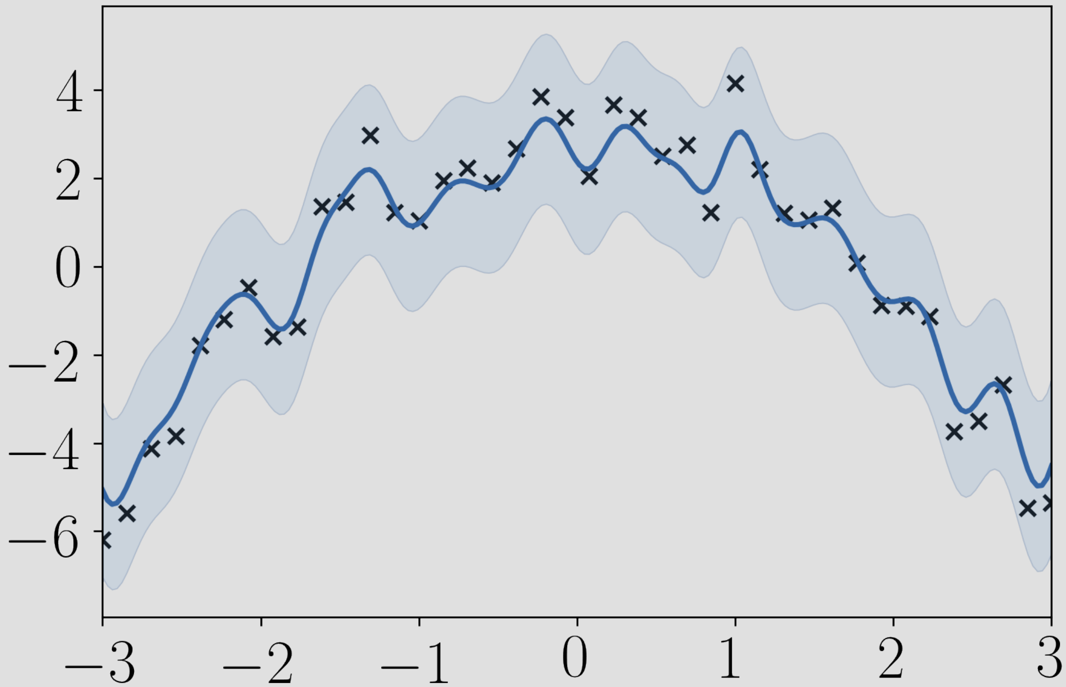
Noisy observations

$$\widehat{f}(x) = f(x) + \epsilon$$ ← Independent Gaussian noise

$$\epsilon \sim \mathcal{N}(0, s^2)$$

$$m(x) = 0$$

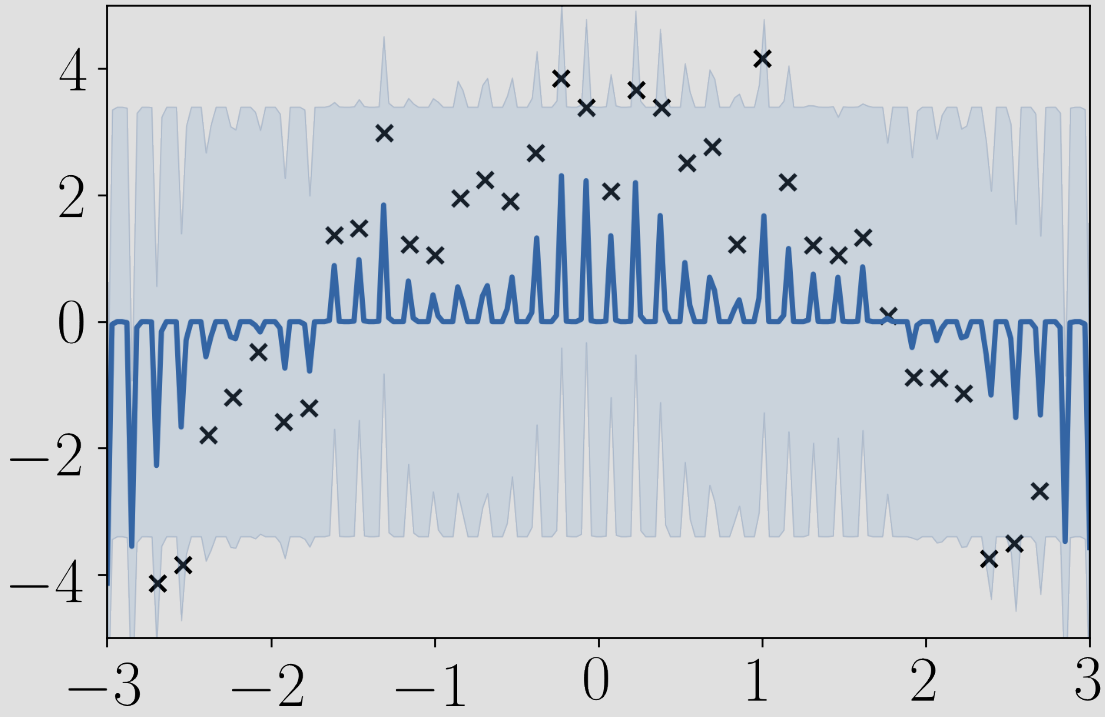
$$\widehat{K}(x_i - x_j) = K(x_i - x_j) + s^2\mathbb{I}[x_i = x_j]$$



Kernel parameters

$$\widehat{K}(x_1 - x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2l^2}\right) + s^2 \mathbb{I}[x_i = x_j]$$

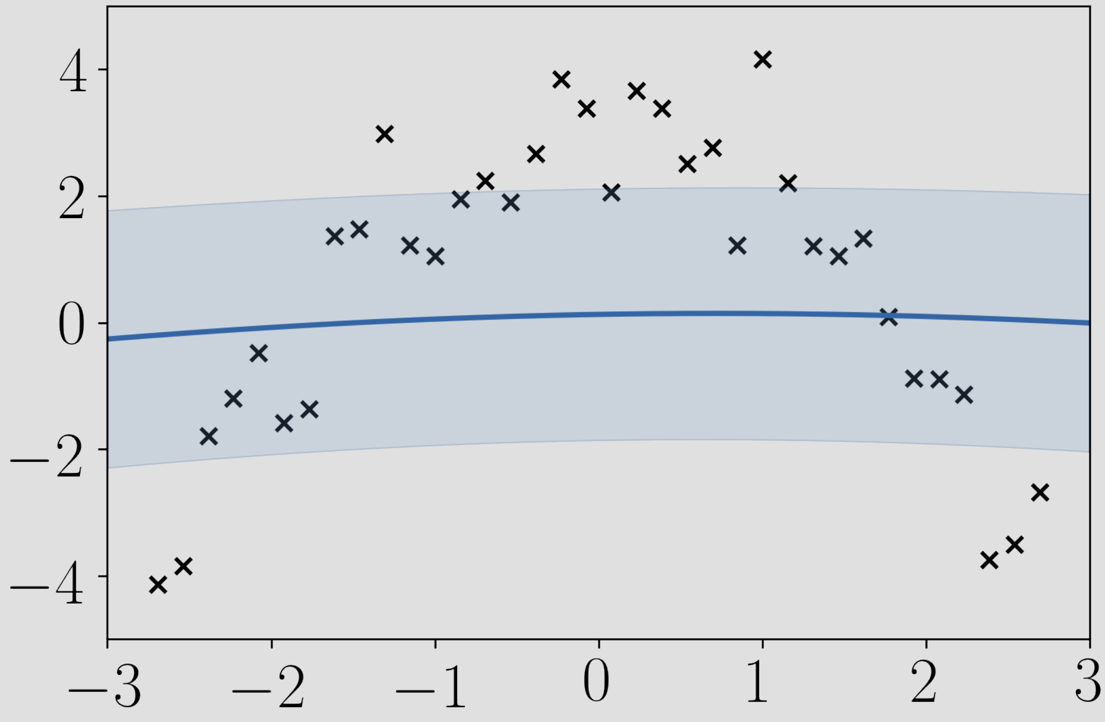
$$l = 0.01$$



Kernel parameters

$$\widehat{K}(x_1 - x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2l^2}\right) + s^2 \mathbb{I}[x_i = x_j]$$

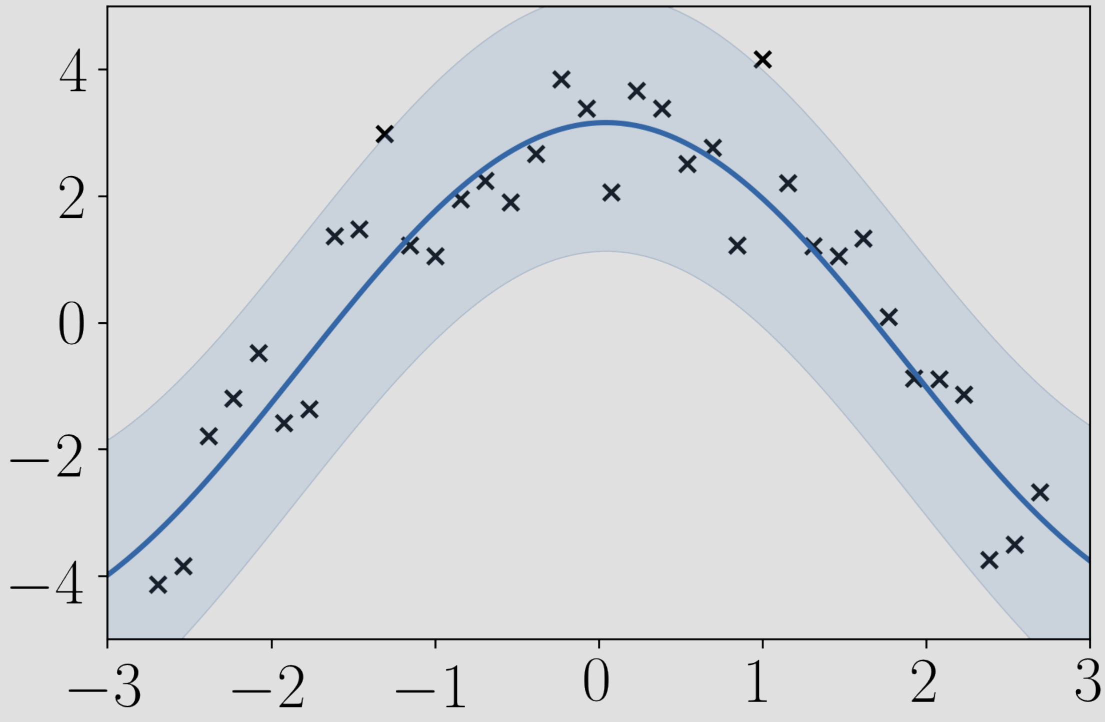
$$l = 10$$



Kernel parameters

$$\widehat{K}(x_1 - x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2l^2}\right) + s^2\mathbb{I}[x_i = x_j]$$

$l = 2$



Kernel parameters

$$\widehat{K}(x_1 - x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2l^2}\right) + s^2 \mathbb{I}[x_i = x_j]$$

$$p(f(x_1), f(x_2), \dots, f(x_n) | \sigma^2, l, s^2) \to \max_{\sigma^2, l, s^2}$$

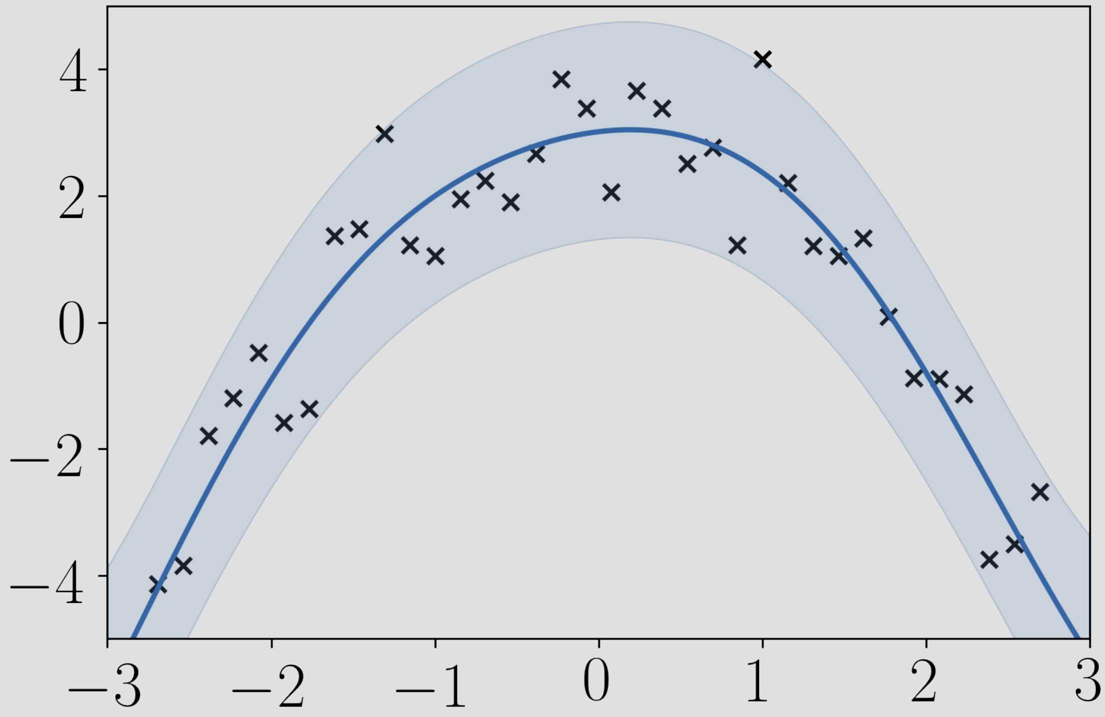
$$\mathcal{N}(f(x_1), f(x_2), \dots, f(x_n) | 0, C) \to \max_{\sigma^2, l, s^2}$$

Optimize with gradient ascent

Kernel parameters

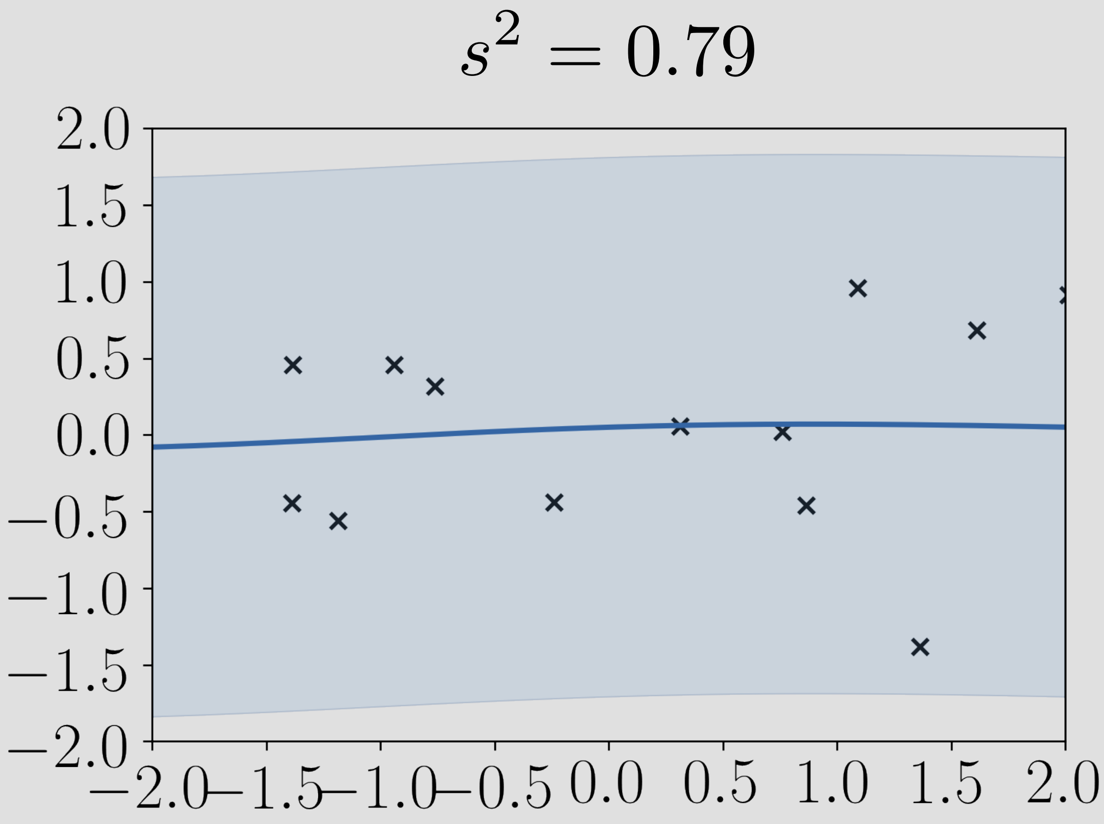
$$\widehat{K}(x_1 - x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2l^2}\right) + s^2 \mathbb{I}[x_i = x_j]$$

$$\sigma^2 = 46.4,\ l = 2,\ s^2 = 0.7$$



Kernel parameters

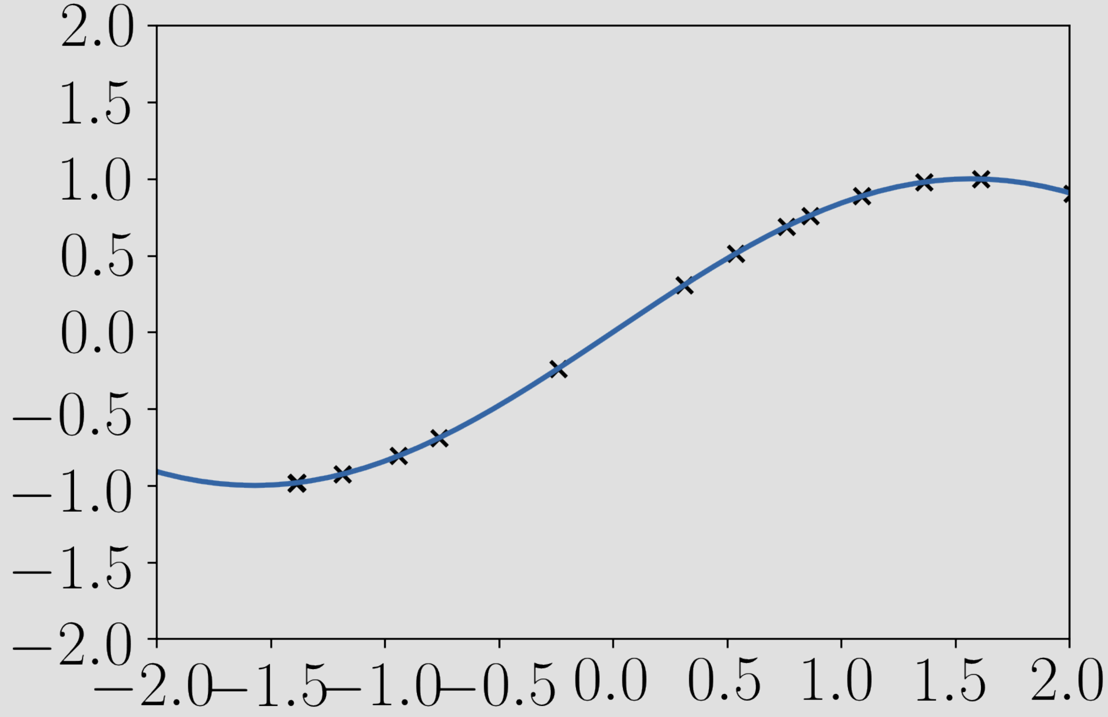
Fitting into **noise**



Kernel parameters

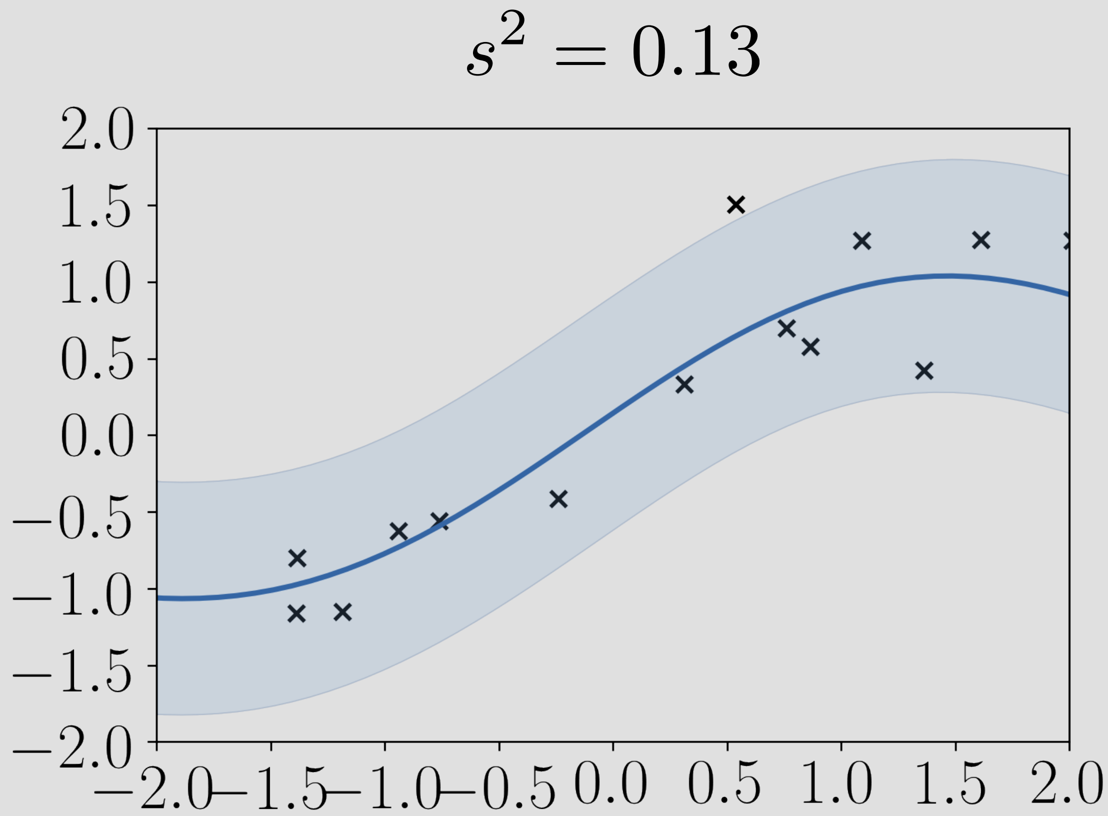
Fitting into signal **without noise**

$$s^2 = 5 \cdot 10^{-17}$$



Kernel parameters

Fitting into **noisy** signal



Classification

$$y \in \{-1, +1\}$$

Latent process: $f(x)$

Class probabilities: $p(y|f) = \dfrac{1}{1 + \exp(-yf)}$

Training:

- Approximate latent process from data

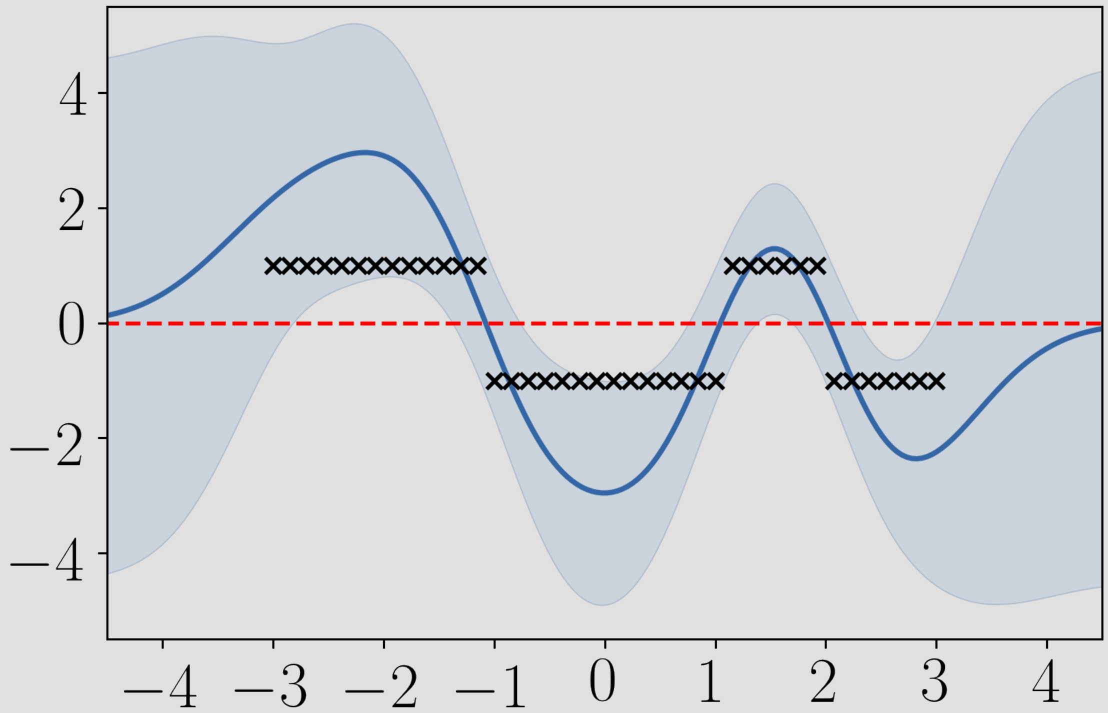
$$p^*(f(x)) = p(f(x) \mid y_1(x_1), \ldots, y_n(x_n))$$

- Compute predictions

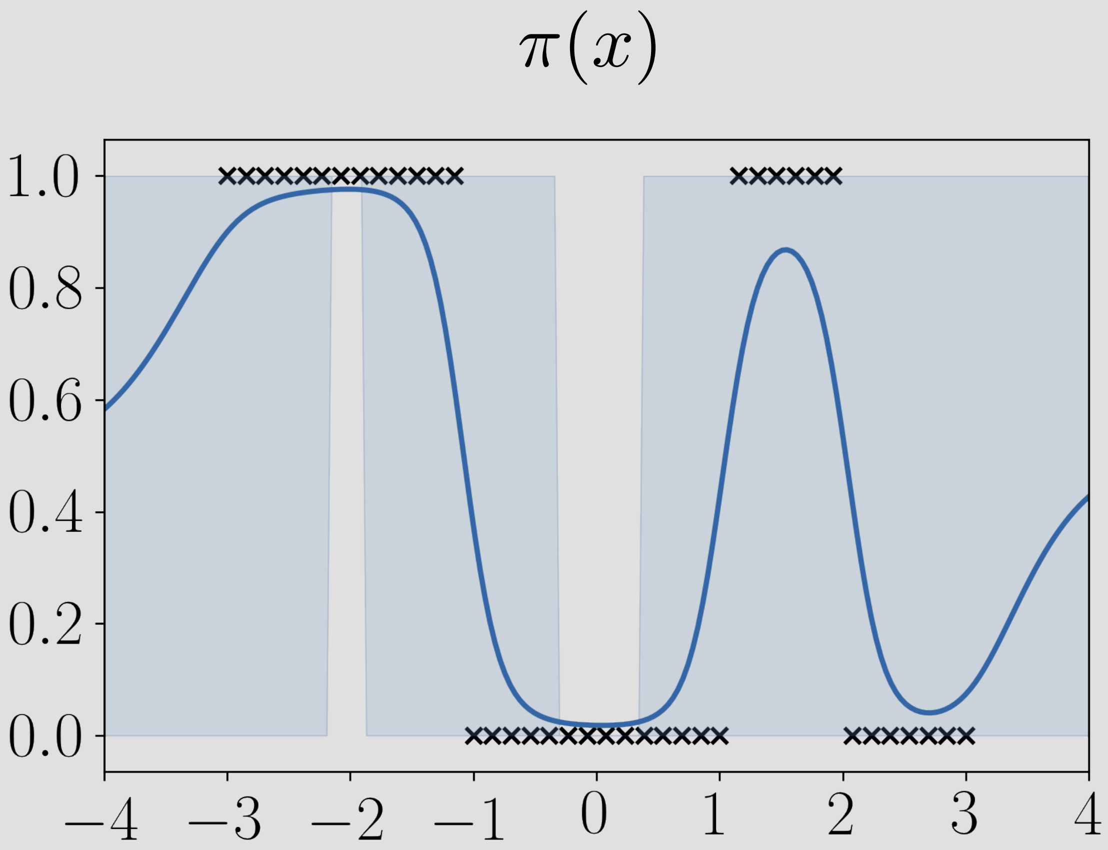
$$\pi(x) = \int p(y(x) \mid f(x)) \cdot p^*(f(x)) df(x)$$

Classification

$$p(f(x) \mid y_1(x_1), \ldots, y_n(x_n))$$



Classification



Inducing inputs

$$\boxed{\begin{aligned} \mu &= k^T C^{-1} f \\ \sigma^2 &= K(0) - k^T C^{-1} k \end{aligned}}$$

$\mathcal{O}(n^3)$

Idea:

- Replace dataset with small number of points (like SVM)
- Predict using those points

Speed:

- Precomputing: $\mathcal{O}(m^2 n)$
- Mean: $\mathcal{O}(m)$
- Variance: $\mathcal{O}(m^2)$

Inducing inputs (ТЕХНИЧЕСКИЙ СЛАЙД)

$$\boxed{\begin{aligned} \mu &= k^T C^{-1} f \\ \sigma^2 &= K(0) - k^T C^{-1} k \end{aligned}}$$

$$\mathcal{O}(n^3)$$

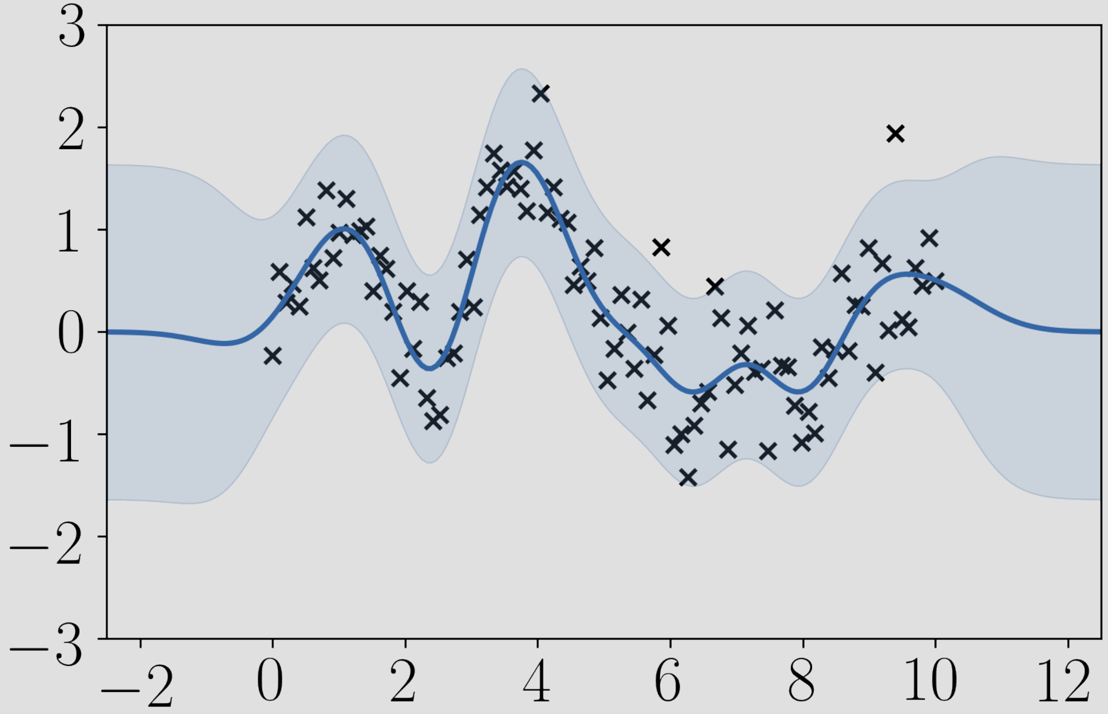
Idea:

- Replace dataset with small number of points (like SVM)
- Predict using those points

Optimize position & values by MLE

Inducing inputs

Full dataset (100 points)



Inducing inputs