Introduction to Gaussian Processes

Jing Zhao
jzhao@cs.ecnu.edu.cn

refer to the slide of Neil D. Lawrence at MLSS

Introducing Gaussian Processes:

- A Gaussian distribution depends on a mean and a covariance vector / matrix.
- A Gaussian process depends on a mean and a covariance function.

Next: Demo, from Gaussian distributions to Gaussian processes.

Gaussian Processes

A Gaussian process defines a distribution over functions, $p(f)$, where f is a function mapping some input space $\mathcal{X}$ to $\Re$.

$$f : \mathcal{X} \to \Re.$$

Notice that f can be an infinite-dimensional quantity (e.g. if $\mathcal{X} = \Re$)

Let $\mathbf{f} = (f(x_1), \ldots, f(x_n))$ be an n-dimensional vector of function values evaluated at n points $x_i \in \mathcal{X}$. Note $\mathbf{f}$ is a random variable.

Definition: $p(f)$ is a Gaussian process if for *any* finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the marginal distribution over that finite subset $p(\mathbf{f})$ has a multivariate Gaussian distribution.

Gaussian process covariance functions (kernels)

$p(f)$ is a Gaussian process if for *any* finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the marginal distribution over that finite subset $p(\mathbf{f})$ has a multivariate Gaussian distribution.

Gaussian processes (GPs) are parameterized by a mean function, $\mu(x)$, and a covariance function, or kernel, $K(x, x')$.

$$p(f(x), f(x')) = \mathsf{N}(\boldsymbol{\mu}, \Sigma)$$

where

$$\boldsymbol{\mu} = \begin{bmatrix} \mu(x) \\ \mu(x') \end{bmatrix} \quad \Sigma = \begin{bmatrix} K(x,x) & K(x,x') \\ K(x',x) & K(x',x') \end{bmatrix}$$

and similarly for $p(f(x_1), \ldots, f(x_n))$ where now $\boldsymbol{\mu}$ is an $n \times 1$ vector and Σ is an $n \times n$ matrix.

Gaussian process covariance functions

Gaussian processes (GPs) are parameterized by a mean function, $\mu(x)$, and a covariance function, $K(x, x')$.

An example covariance function:

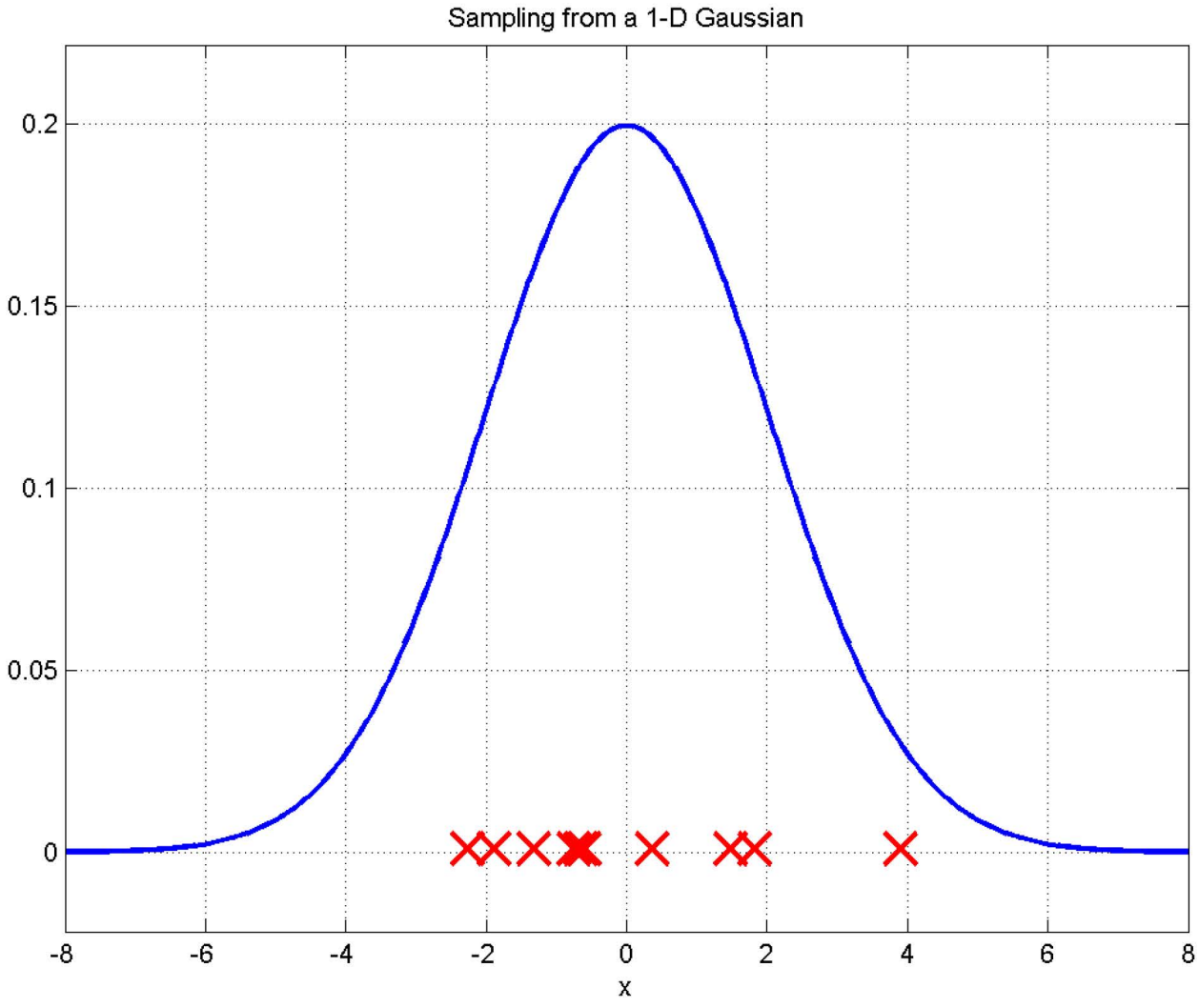
$$K(x_i, x_j) = v_0 \exp\left\{-\left(\frac{|x_i - x_j|}{r}\right)^{\alpha}\right\} + v_1 + v_2\,\delta_{ij}$$

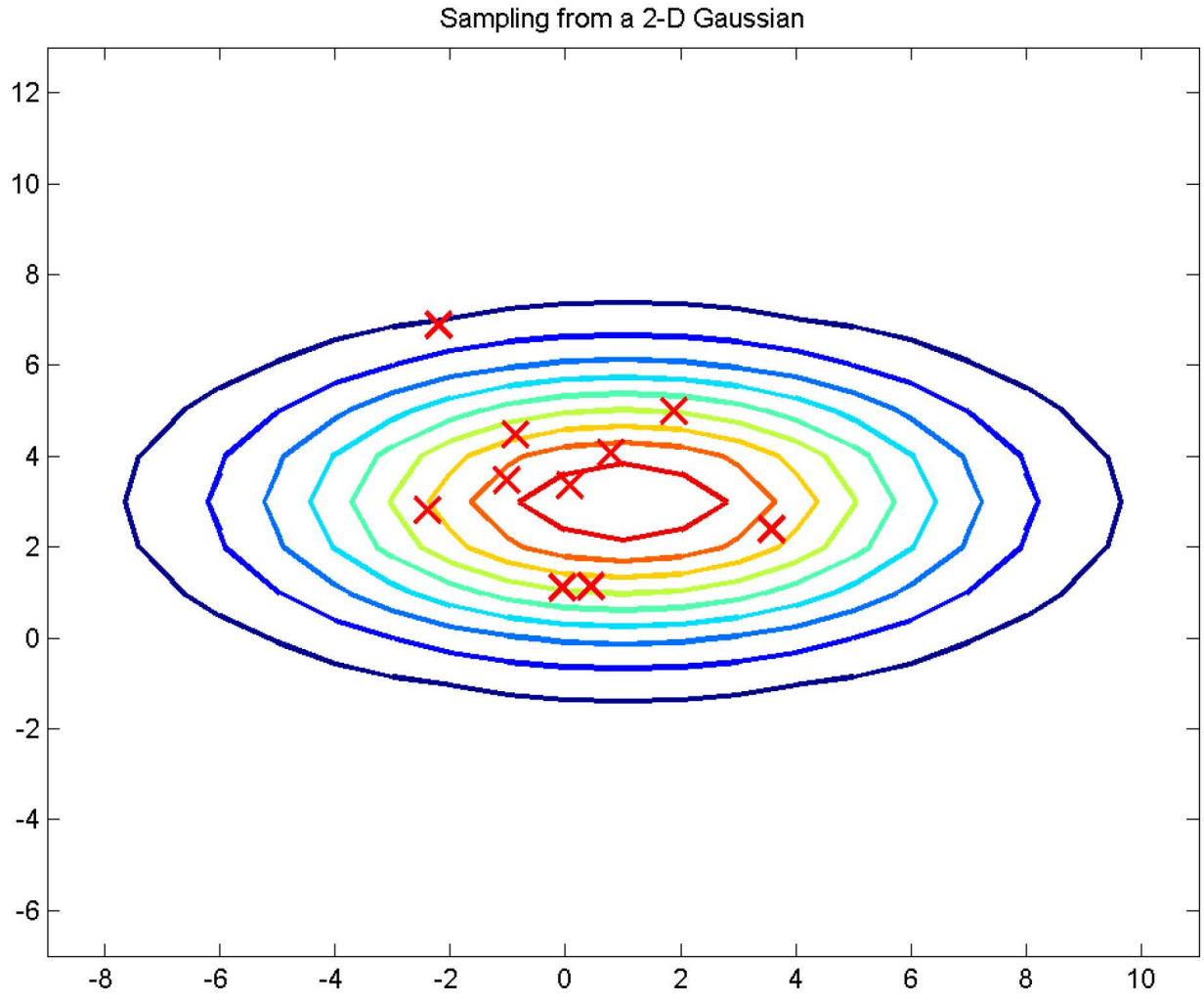
with parameters $(v_0, v_1, v_2, r, \alpha)$

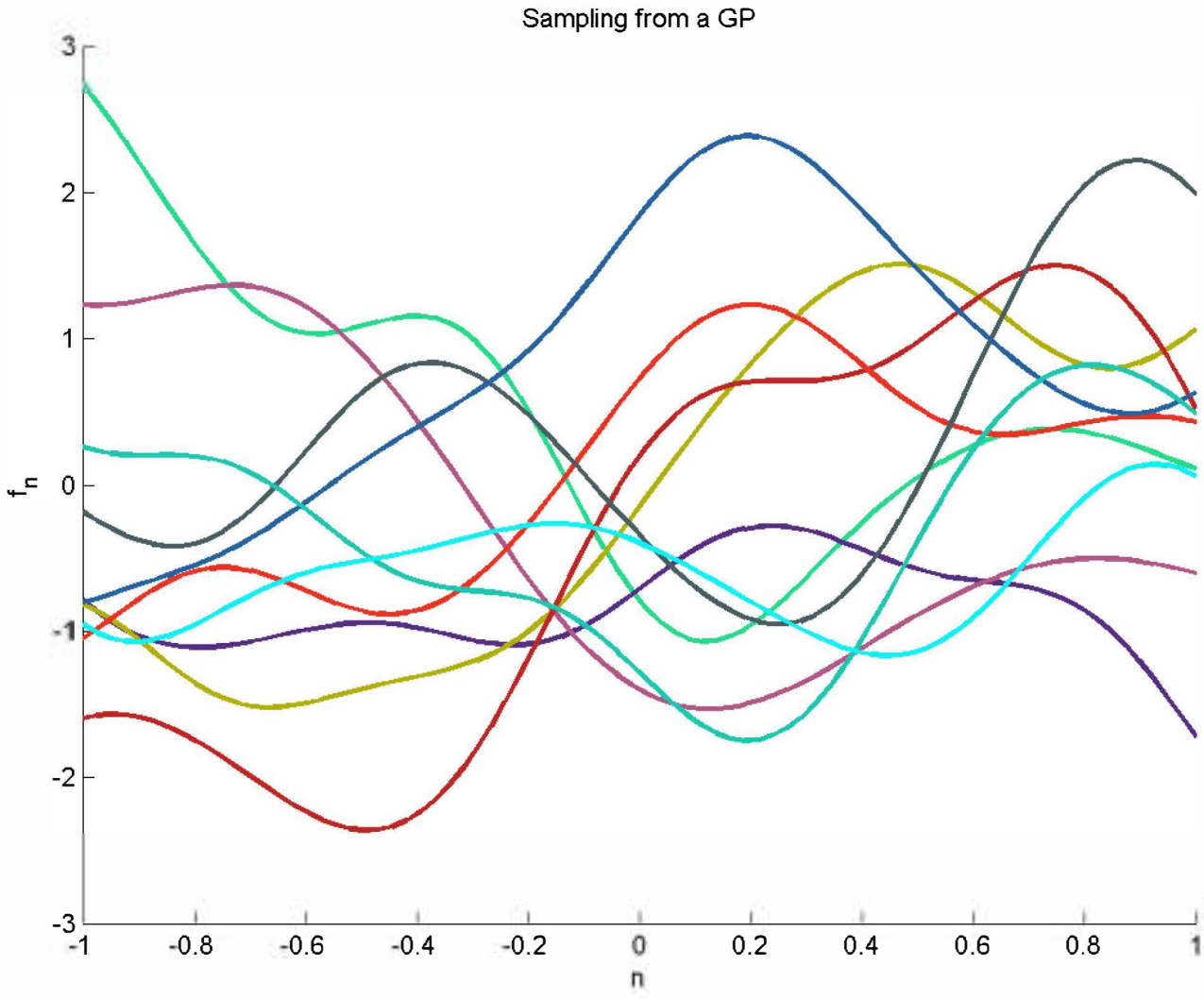
These kernel parameters are interpretable and can be learned from data:

v_0	signal variance
v_1	variance of bias
v_2	noise variance
r	lengthscale
α	roughness

Once the mean and covariance functions are defined, everything else about GPs follows from the basic rules of probability applied to mutivariate Gaussians.


Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x


Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10


Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

Infinite model... but we *always* work with finite sets!

Let's start with a multivariate Gaussian:

$$p(\underbrace{f_1, f_2, \cdots, f_s}_{\mathbf{f}_A}, \underbrace{f_{s+1}, f_{s+2}, \cdots, f_N}_{\mathbf{f}_B}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$$

$$\text{OR:} \quad p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}).$$

with:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Marginalisation property:

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Infinite model... but we *always* work with finite sets!

Let's start with a multivariate Gaussian:

$$p(\underbrace{f_1, f_2, \cdots, f_s}_{\mathbf{f}_A}, \underbrace{f_{s+1}, f_{s+2}, \cdots, f_N}_{\mathbf{f}_B}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$$

$$\text{OR:} \quad p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}).$$

with:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Marginalisation property:

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Infinite model... but we *always* work with finite sets!

In the GP context:

$$\boldsymbol{\mu}_\infty = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{x}} \\ \cdots \\ \cdots \end{bmatrix} \quad \text{and} \quad \mathbf{K}_\infty = \begin{bmatrix} \mathbf{K}_{\mathbf{xx}} & \cdots \\ \cdots & \cdots \end{bmatrix}$$

where:

Training data:

$$\begin{aligned} \mathbf{X} &= [x_1, \cdots, x_N] \\ \mathbf{f} &= [f_1, \cdots, f_N] = [f(x_1), \cdots, f(x_N)] \end{aligned}$$

Posterior is also Gaussian!

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$
$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\cdots, \cdots)$$

In the GP context this can be used for inter/extrapolation:

$$p(f_* | f_1, \cdots, f_N) = p(f(x_*) | f(x_1), \cdots, f(x_N)) \sim \mathcal{N}$$

Posterior is also Gaussian!

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$
$$p(\mathbf{f}_A|\mathbf{f}_B) = \mathcal{N}(\cdots, \cdots)$$

In the GP context this can be used for inter/extrapolation:

$$p(f_*|f_1, \cdots, f_N) = p(f(x_*)|f(x_1), \cdots, f(x_N)) \sim \mathcal{N}$$

More about the GP posterior

- For test points $\mathbf{X}_*$ we can predict their values $\mathbf{f}_*$.
- Assuming a zero-mean GP prior, $\mathbf{f}$ and $\mathbf{f}_*$ follow a joint Gaussian:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{XX}} & \mathbf{K}_{\mathbf{XX}_*} \\ \mathbf{K}_{\mathbf{X}_*\mathbf{X}} & \mathbf{K}_{\mathbf{X}_*\mathbf{X}_*} \end{bmatrix}\right)$$

- The conditional $p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*)$ is Gaussian with:

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{K}_{\mathbf{X}_*\mathbf{X}} \mathbf{K}_{\mathbf{XX}}^{-1} \mathbf{f} \\ \boldsymbol{\Sigma} &= \mathbf{K}_{\mathbf{X}_*\mathbf{X}_*} - \mathbf{K}_{\mathbf{X}_*\mathbf{X}} \mathbf{K}_{\mathbf{XX}}^{-1} \mathbf{K}_{\mathbf{XX}_*} \end{aligned}$$

- But where is $\mathbf{K}_{..}$ coming from?

More about the GP posterior

- For test points $\mathbf{X}_*$ we can predict their values $\mathbf{f}_*$.
- Assuming a zero-mean GP prior, $\mathbf{f}$ and $\mathbf{f}_*$ follow a joint Gaussian:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{xx}} & \mathbf{K}_{\mathbf{xx}_*} \\ \mathbf{K}_{\mathbf{x}_*\mathbf{x}} & \mathbf{K}_{\mathbf{x}_*\mathbf{x}_*} \end{bmatrix}\right)$$

- The conditional $p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*)$ is Gaussian with:

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{K}_{\mathbf{x}_*\mathbf{x}} \mathbf{K}_{\mathbf{xx}}^{-1} \mathbf{f} \\ \boldsymbol{\Sigma} &= \mathbf{K}_{\mathbf{x}_*\mathbf{x}_*} - \mathbf{K}_{\mathbf{x}_*\mathbf{x}} \mathbf{K}_{\mathbf{xx}}^{-1} \mathbf{K}_{\mathbf{xx}_*} \end{aligned}$$

- But where is $\mathbf{K}_{..}$ coming from?

Covariance functions

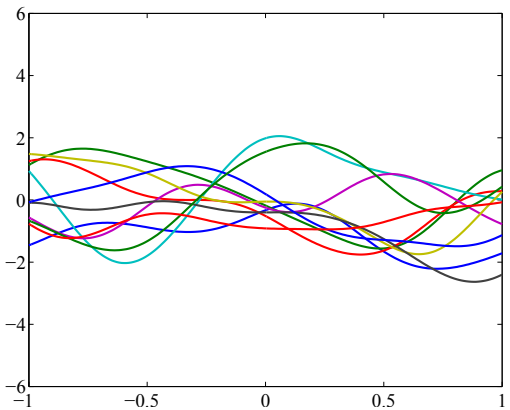
- Assumptions about *properties* of $f \Rightarrow$ define a parametric form for k, e.g:

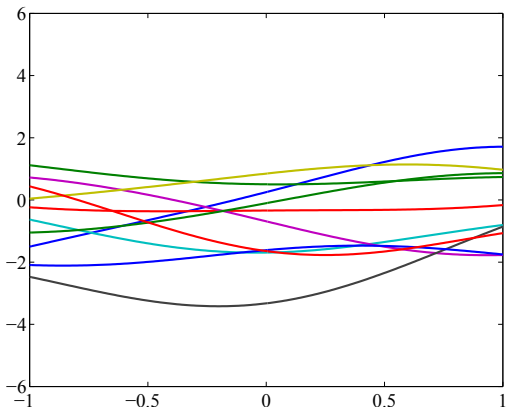
$$k(x, x') = \alpha \exp\left(-\frac{\gamma}{2}(x - x')^\top (x - x')\right)$$

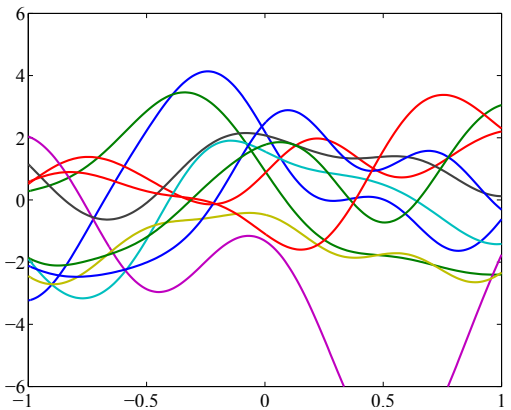
- However, a GP prior with this cov. function defines a whole *family* of functions
- The parameters $\{\alpha, \gamma\}$ are *hyperparameters*.
- We write: $f \sim \mathcal{GP}(0, k(x, x'))$

Covariance samples and hyperparameters

- The hyperparameters of the cov. function define the properties (and NOT an explicit form) of the sampled functions







Incorporating Gaussian noise is tractable

- So far we assumed: $\mathbf{f} = f(\mathbf{X})$
- Assuming that we only observe noisy versions $\mathbf{y}$ of the true outputs $\mathbf{f}$:

$$\mathbf{y} = f(\mathbf{X}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Using Gaussian processes for nonlinear regression

Imagine observing a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^n\} = (\mathbf{X}, \mathbf{y})$.

Model:

$$\begin{aligned} y_i &= f(\mathbf{x}_i) + \epsilon_i \\ f &\sim \mathsf{GP}(\cdot|0, K) \\ \epsilon_i &\sim \mathsf{N}(\cdot|0, \sigma^2) \end{aligned}$$

Prior on f is a GP, likelihood is Gaussian, therefore posterior on f is also a GP.

We can use this to make predictions

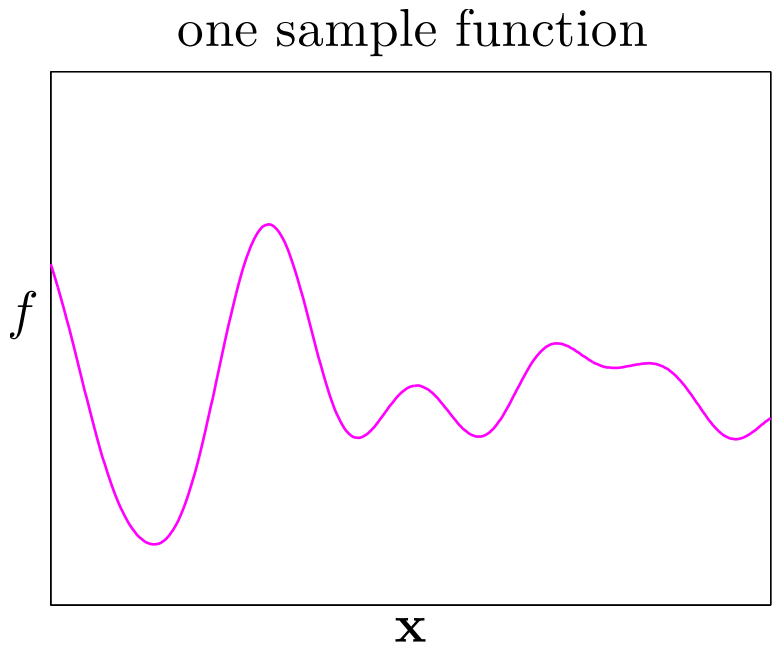
$$p(y_*|\mathbf{x}_*, \mathcal{D}) = \int p(y_*|\mathbf{x}_*, f, \mathcal{D})\, p(f|\mathcal{D})\, df$$

We can also compute the marginal likelihood (evidence) and use this to compare or tune covariance functions

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

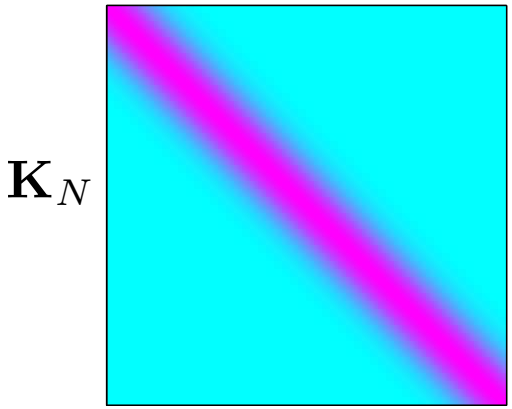
Gaussian process (GP) priors

GP: consistent Gaussian prior on any set of function values $\mathbf{f} = \{f_n\}_{n=1}^N$, given corresponding inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$



prior

$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N)$

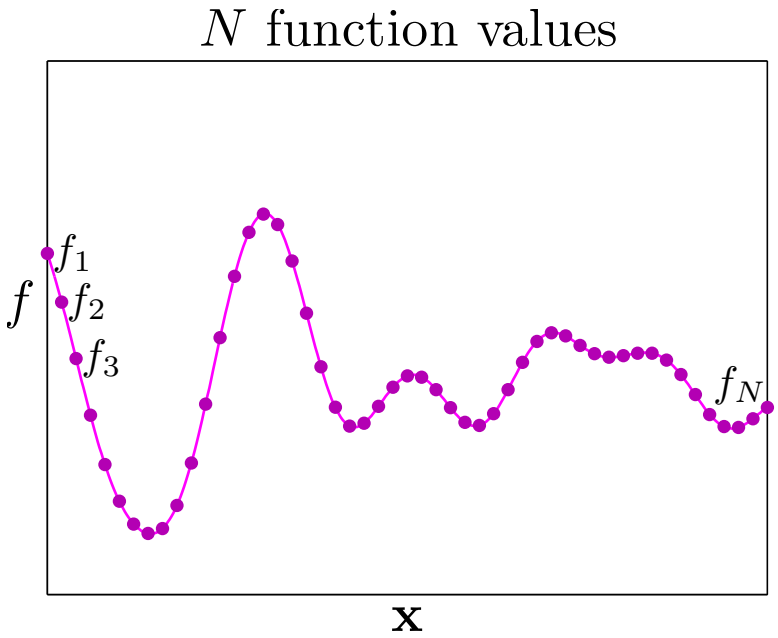


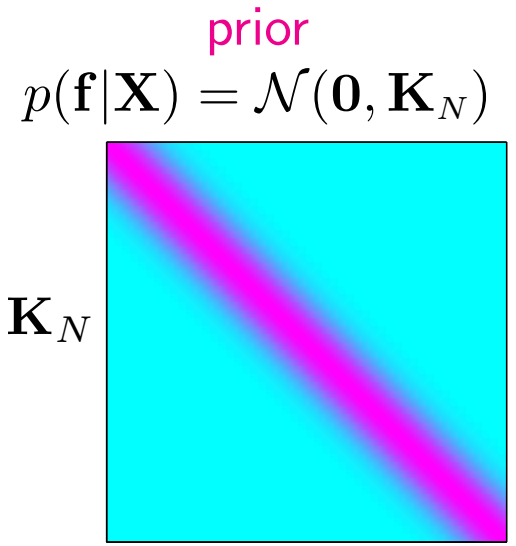
Covariance: $\mathbf{K}_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\theta})$, hyperparameters $\boldsymbol{\theta}$

$$\mathbf{K}_{nn'} = v \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_n^{(d)} - x_{n'}^{(d)}}{r_d}\right)^2\right]$$

Gaussian process (GP) priors

GP: consistent Gaussian prior on any set of function values $\mathbf{f} = \{f_n\}_{n=1}^N$, given corresponding inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$



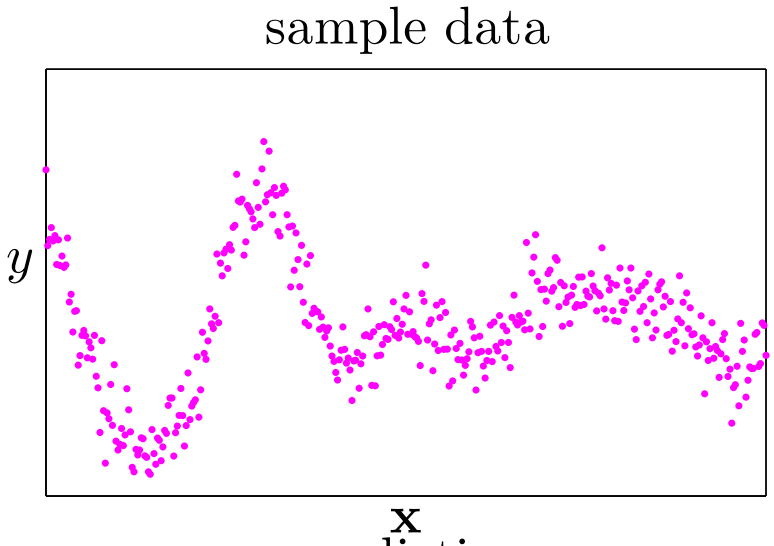


Covariance: $\mathbf{K}_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\theta})$, hyperparameters $\boldsymbol{\theta}$

$$\mathbf{K}_{nn'} = v \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_n^{(d)} - x_{n'}^{(d)}}{r_d}\right)^2\right]$$

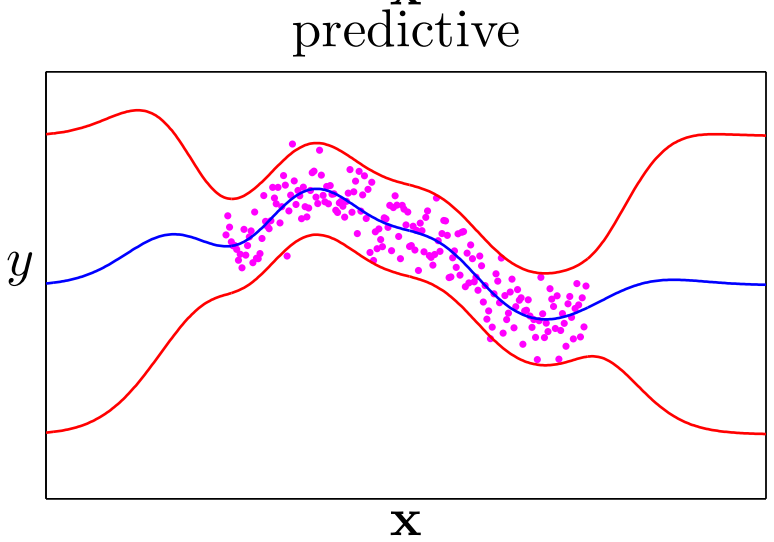
GP regression

Gaussian observation noise: $y_n = f_n + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$



marginal likelihood

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N + \sigma^2\mathbf{I})$$



predictive distribution

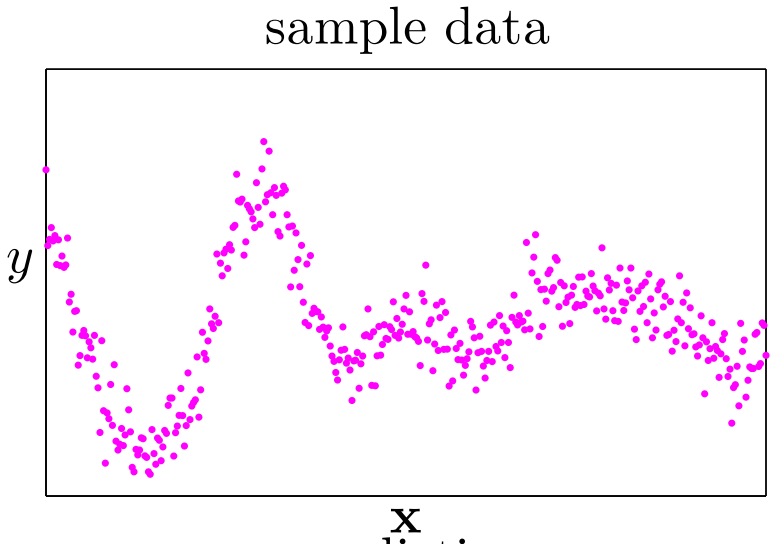
$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \sigma_*^2)$$

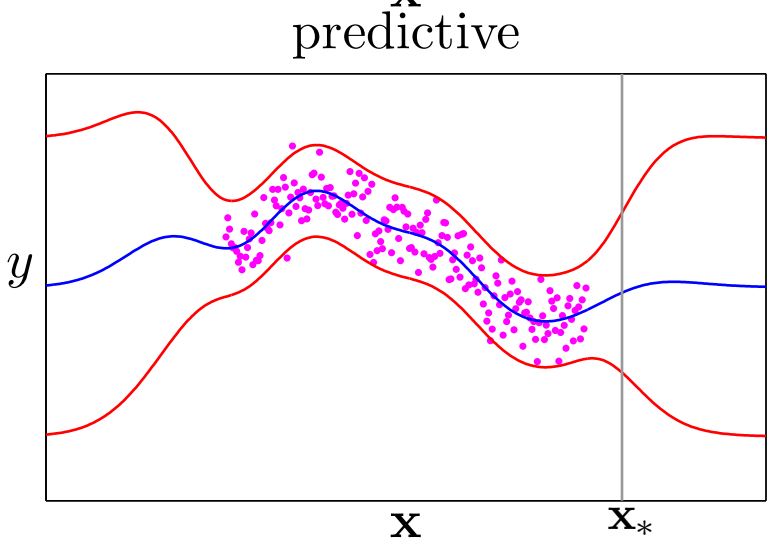
$$\mu_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\sigma_*^2 = K_{**} - \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{N*} + \sigma^2$$

GP regression

Gaussian observation noise: $y_n = f_n + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$





marginal likelihood

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N + \sigma^2\mathbf{I})$$

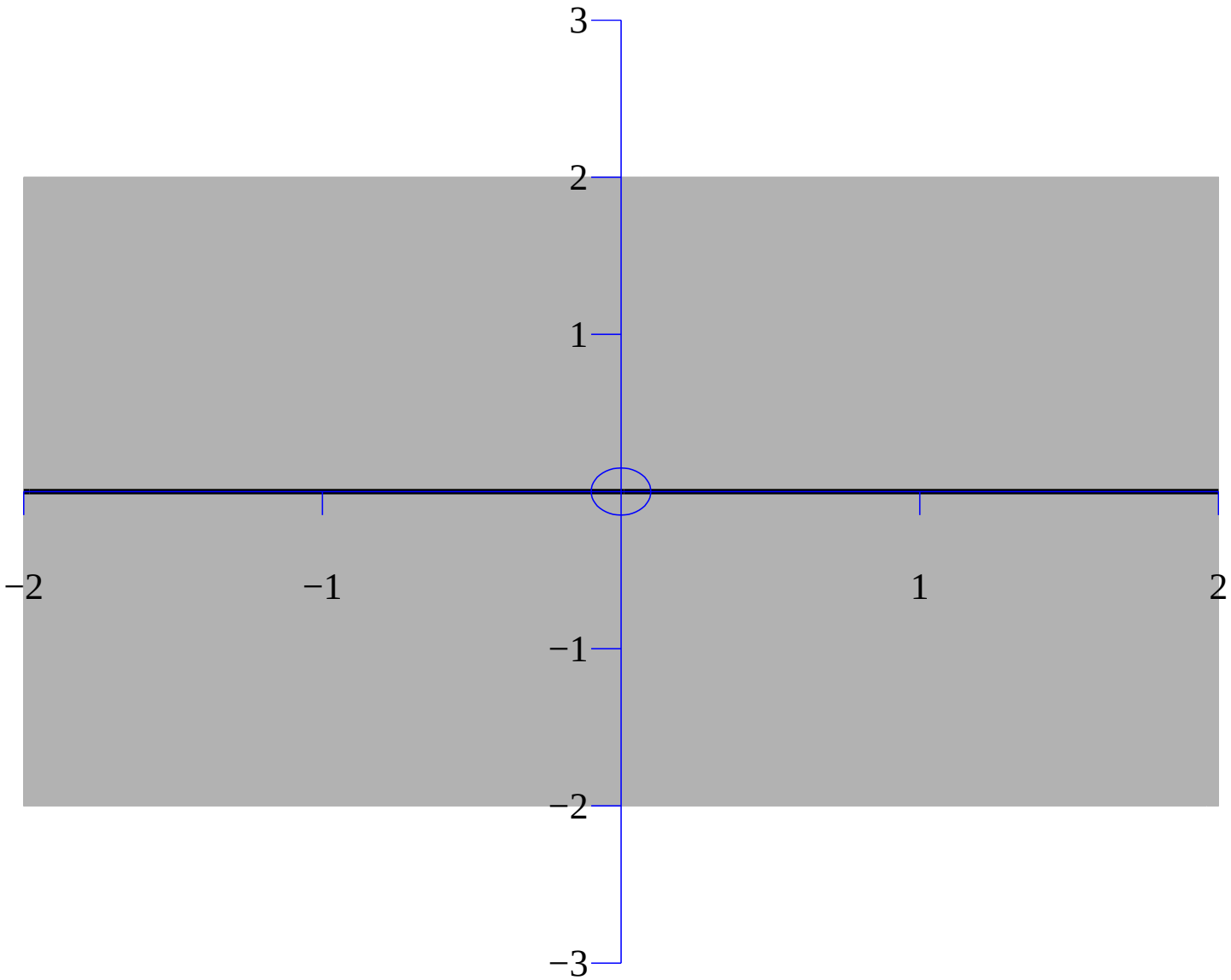
predictive distribution

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \sigma_*^2)$$

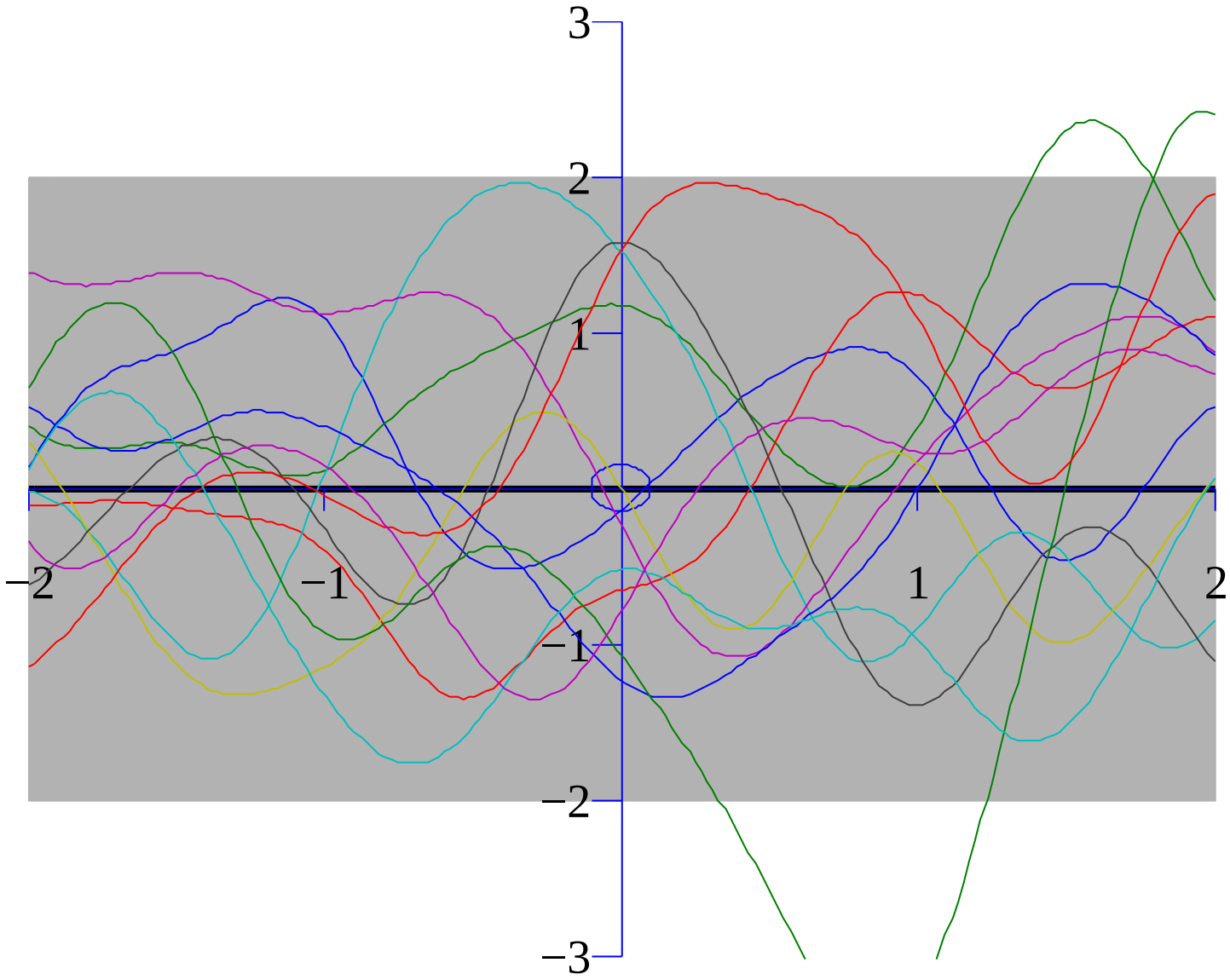
$$\mu_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\sigma_*^2 = K_{**} - \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{N*} + \sigma^2$$

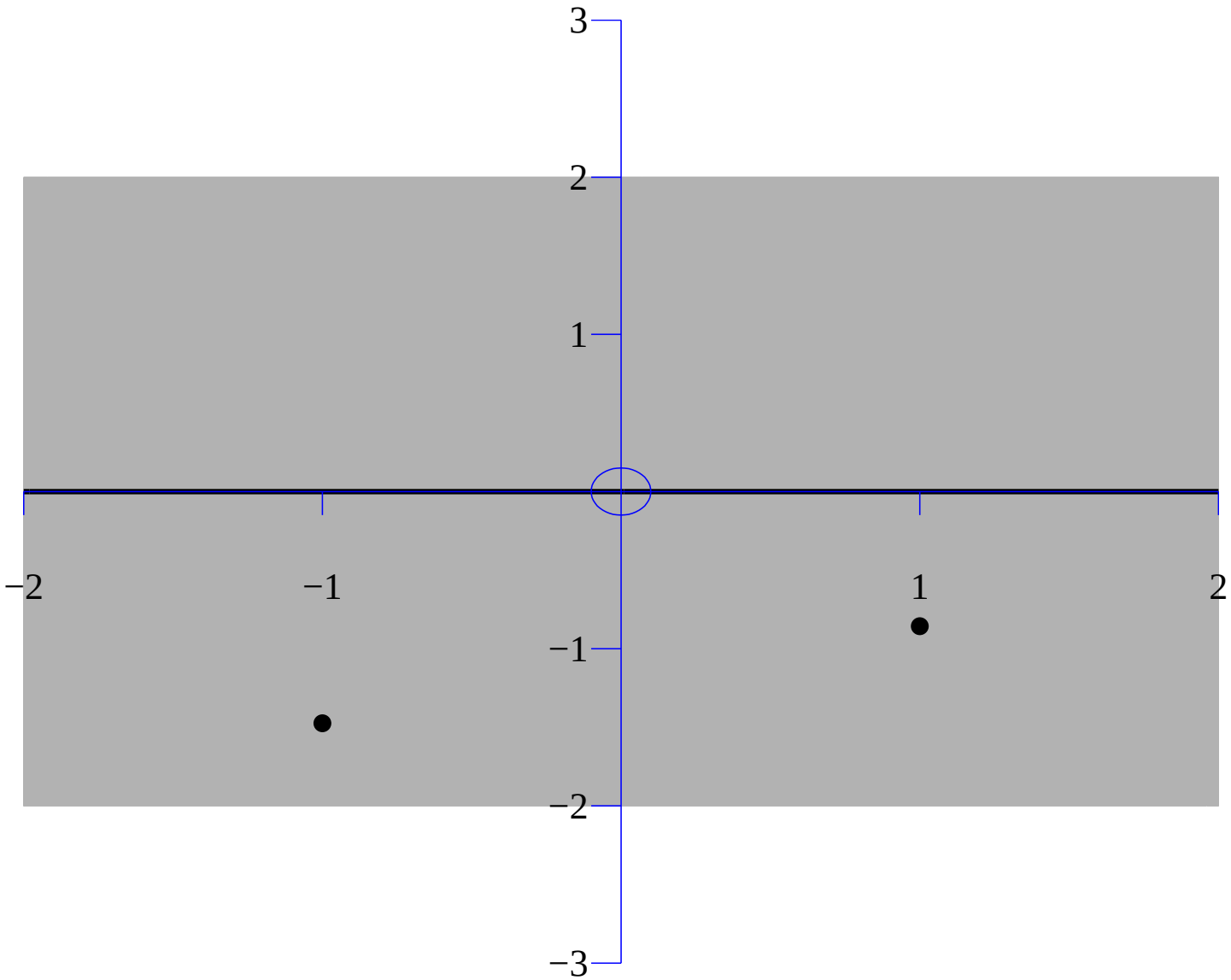
Fitting the data



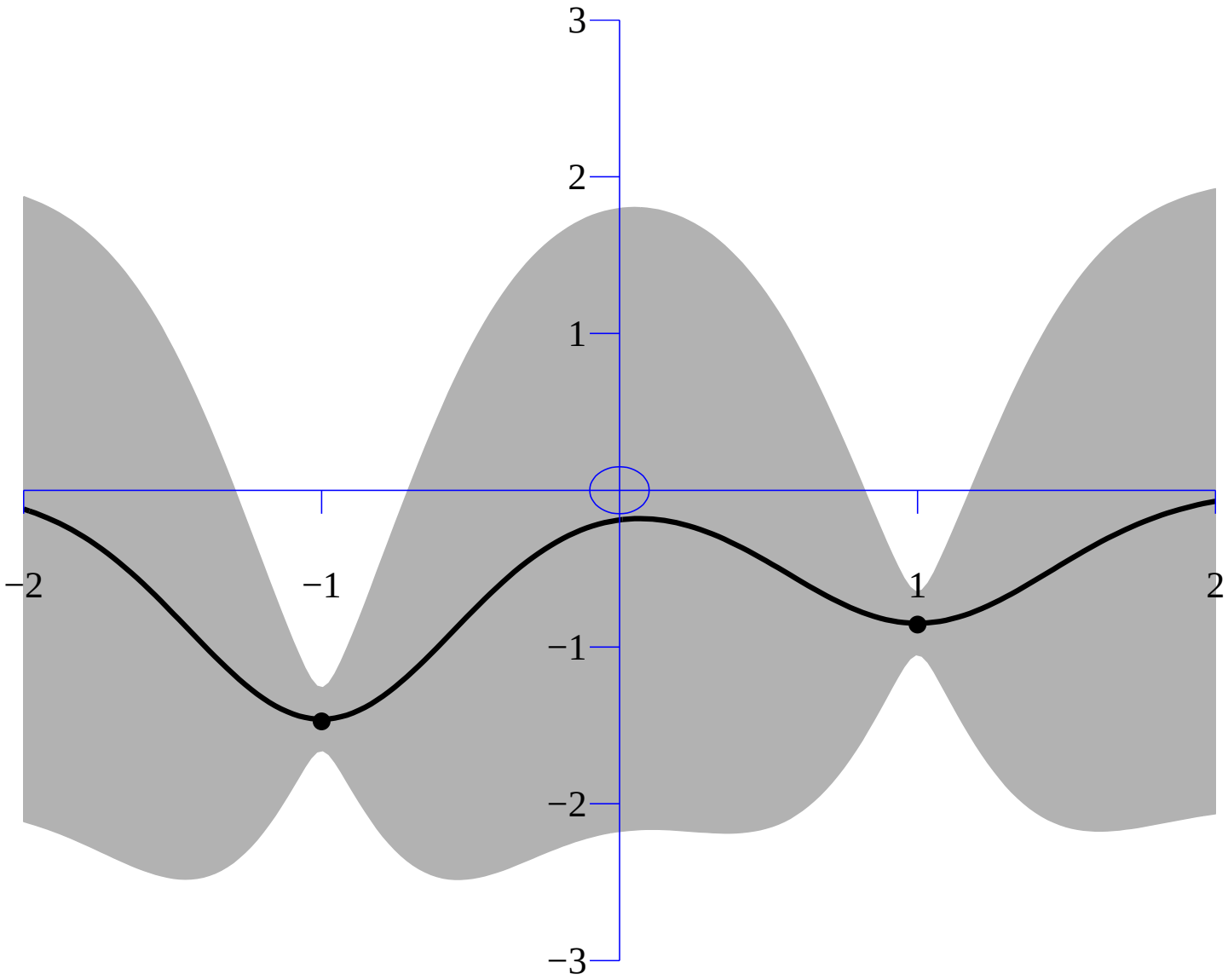
Fitting the data - Prior Samples



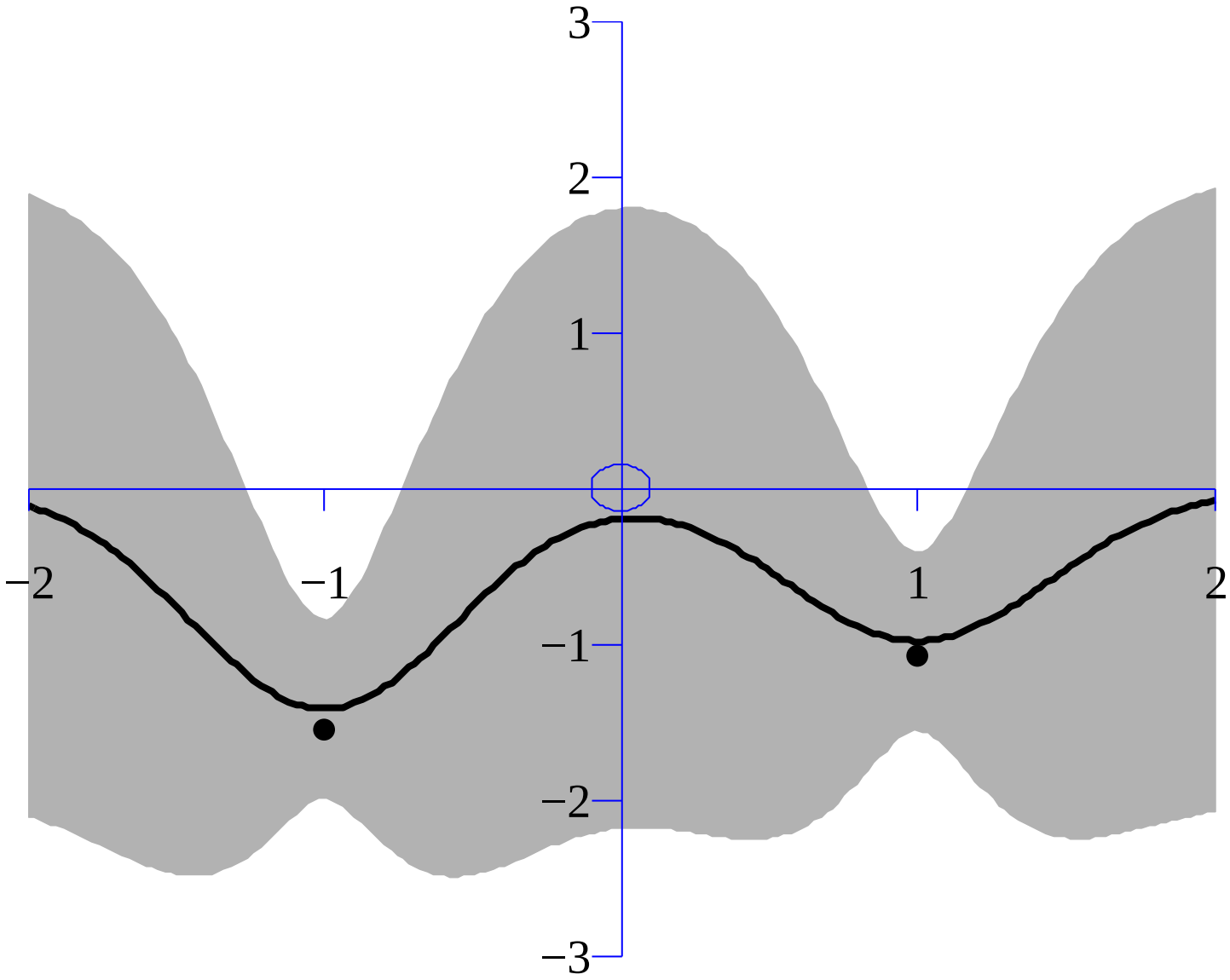
Fitting the data



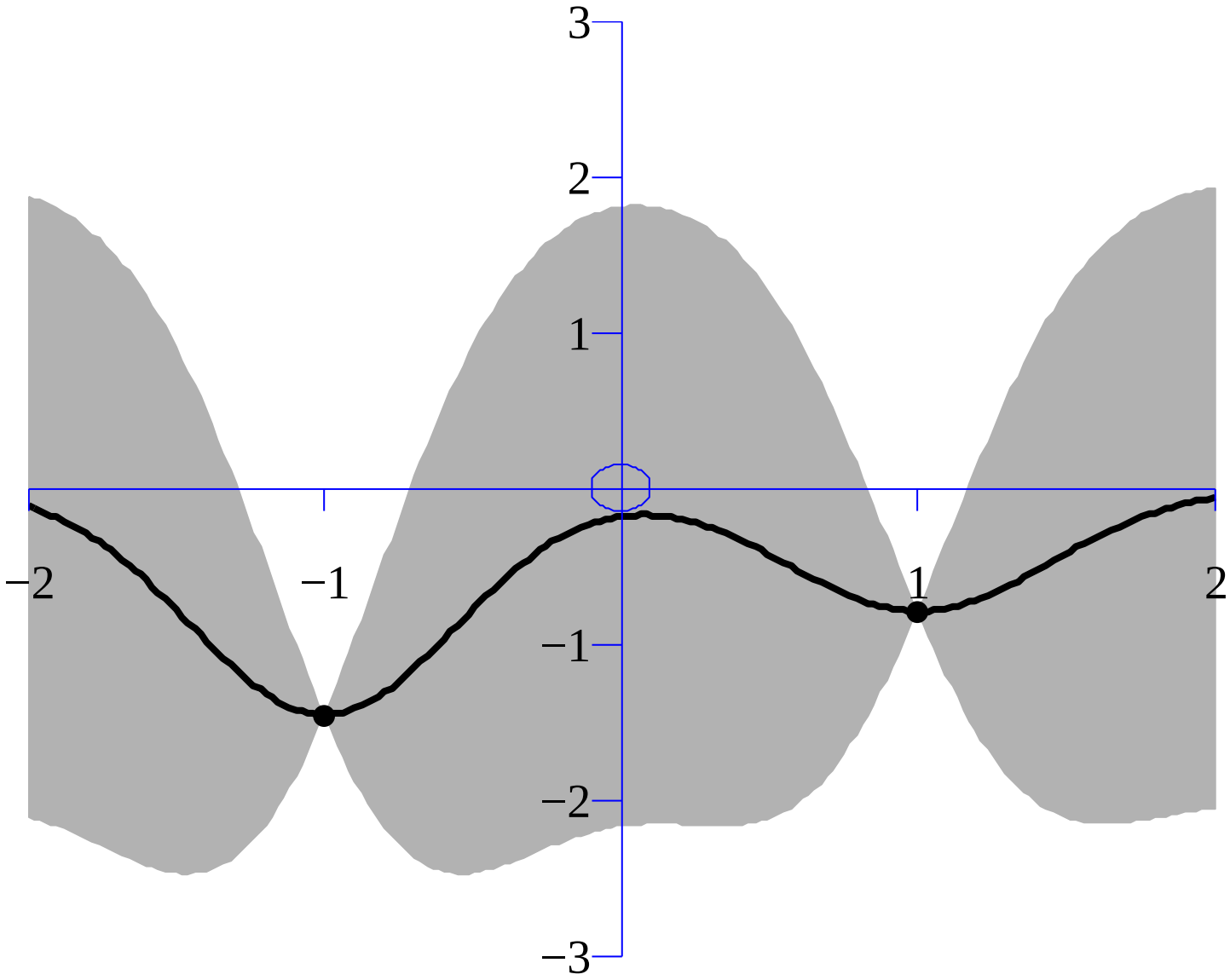
Fitting the data



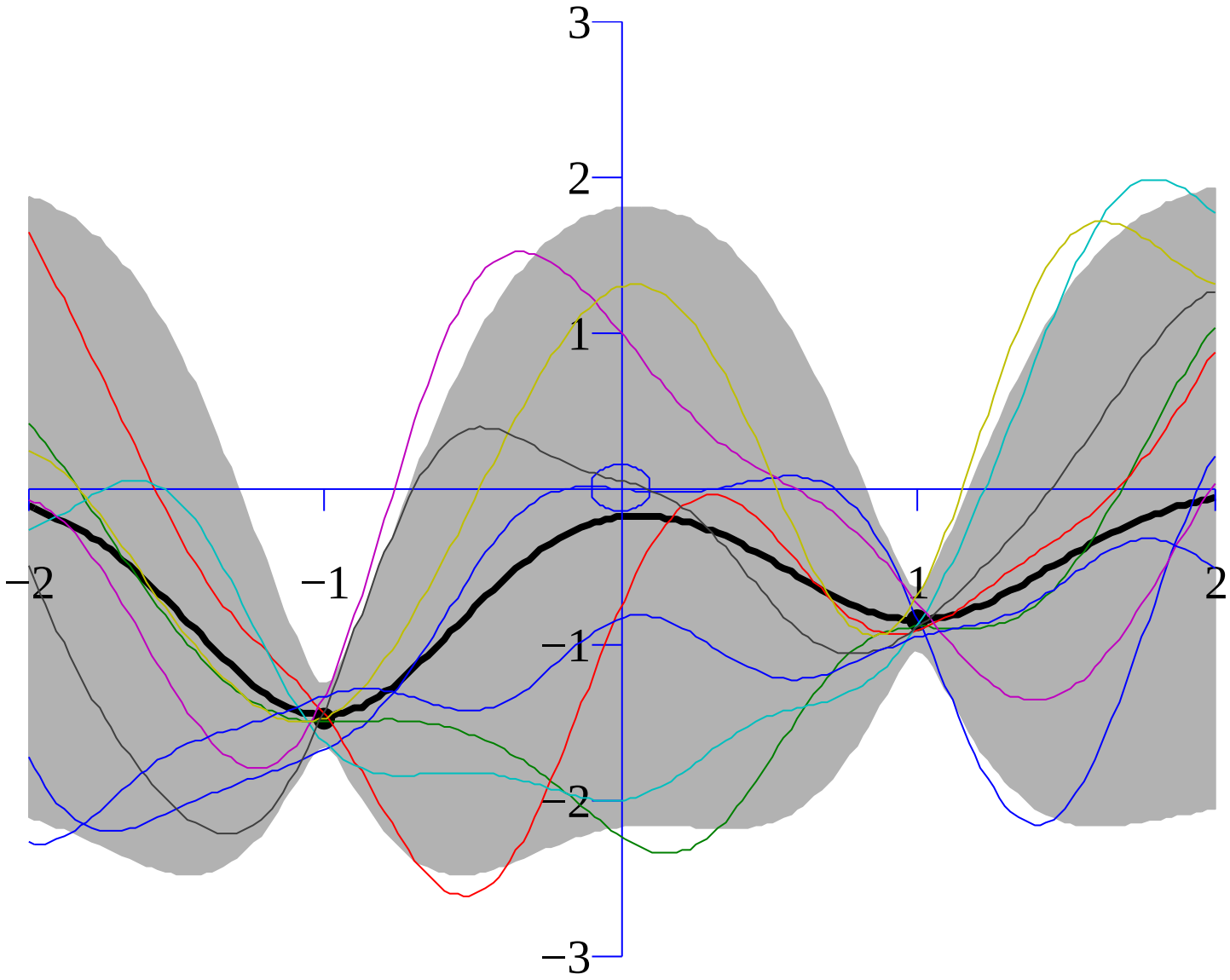
Fitting the data - more noise



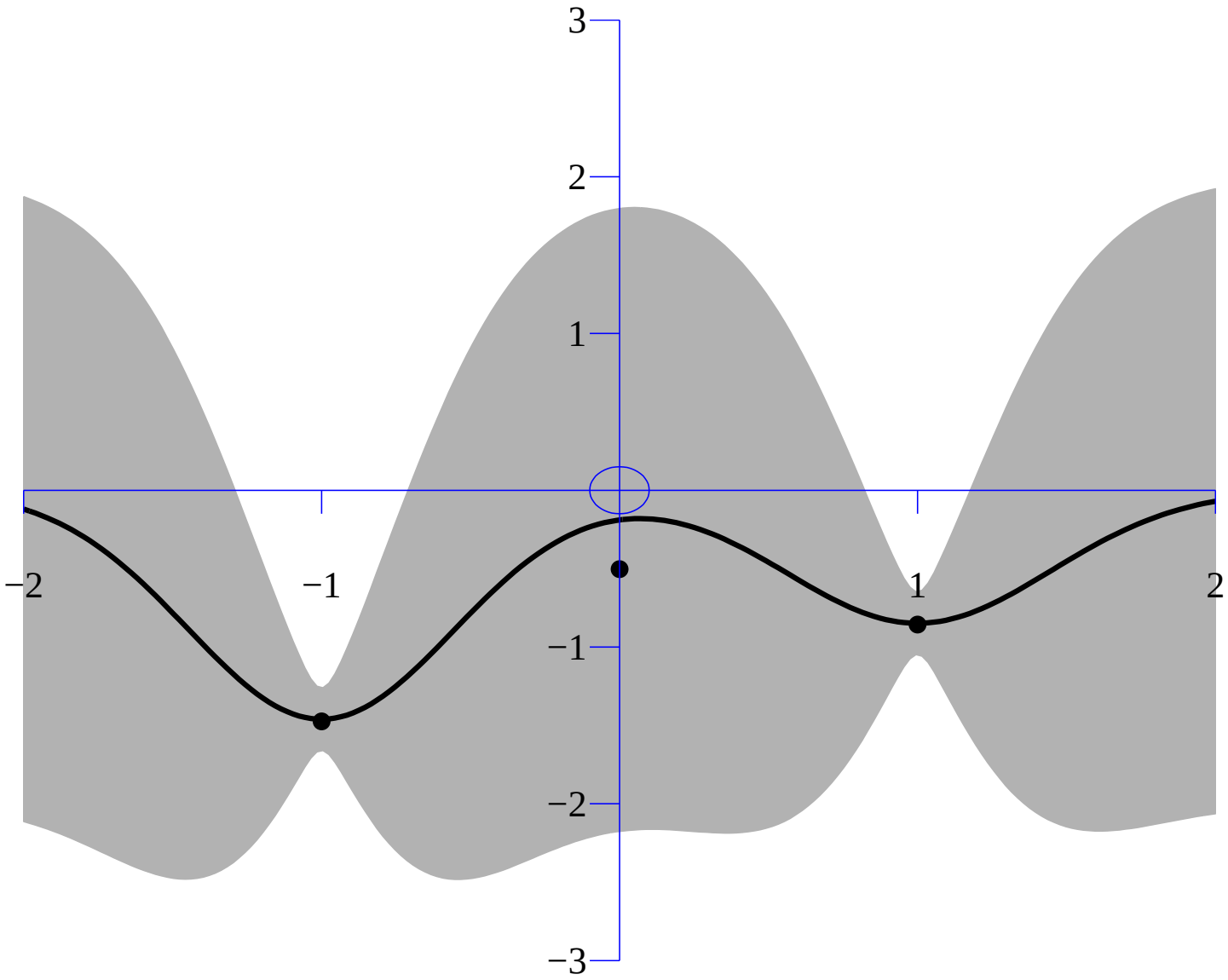
Fitting the data - no noise



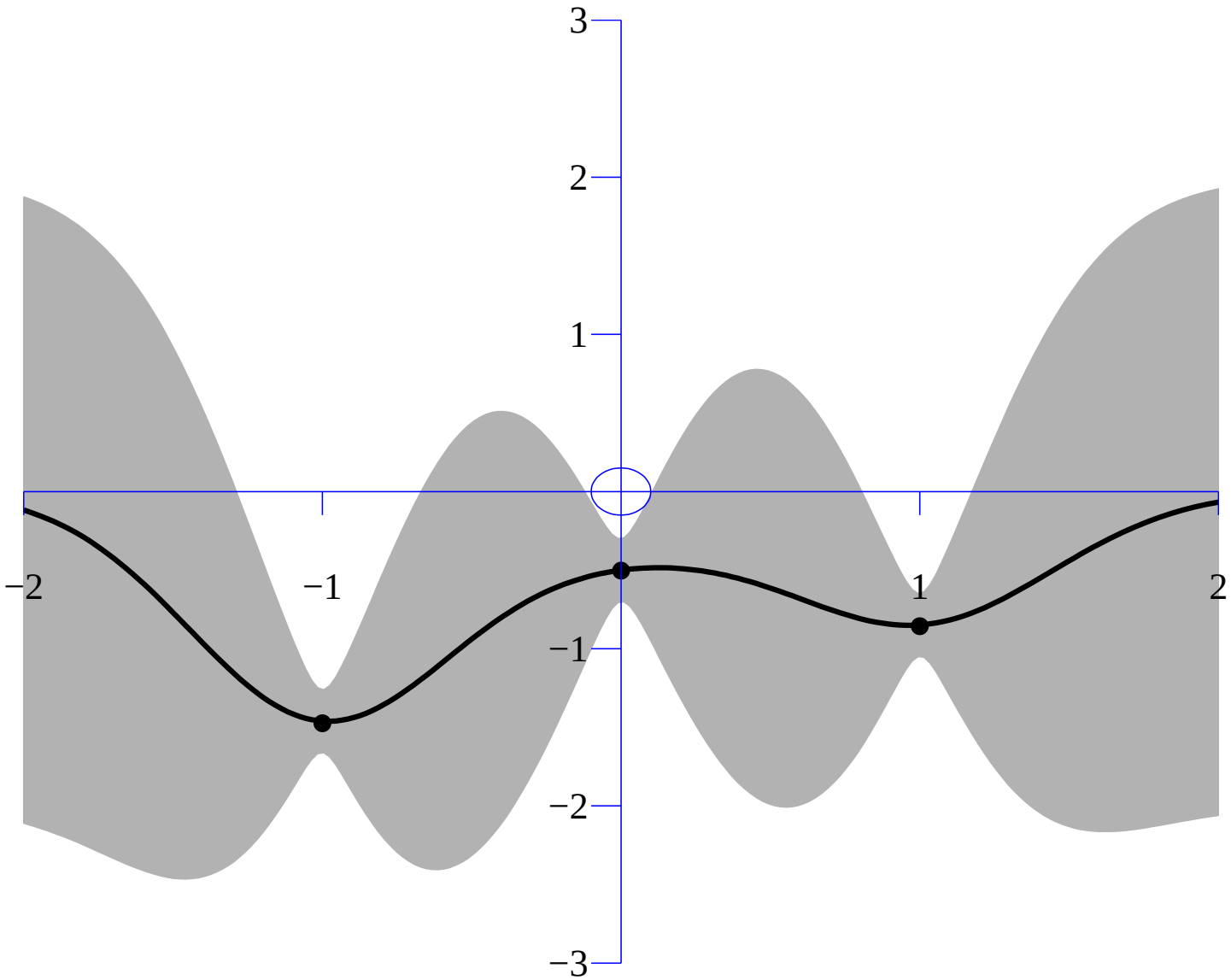
Fitting the data - Posterior samples



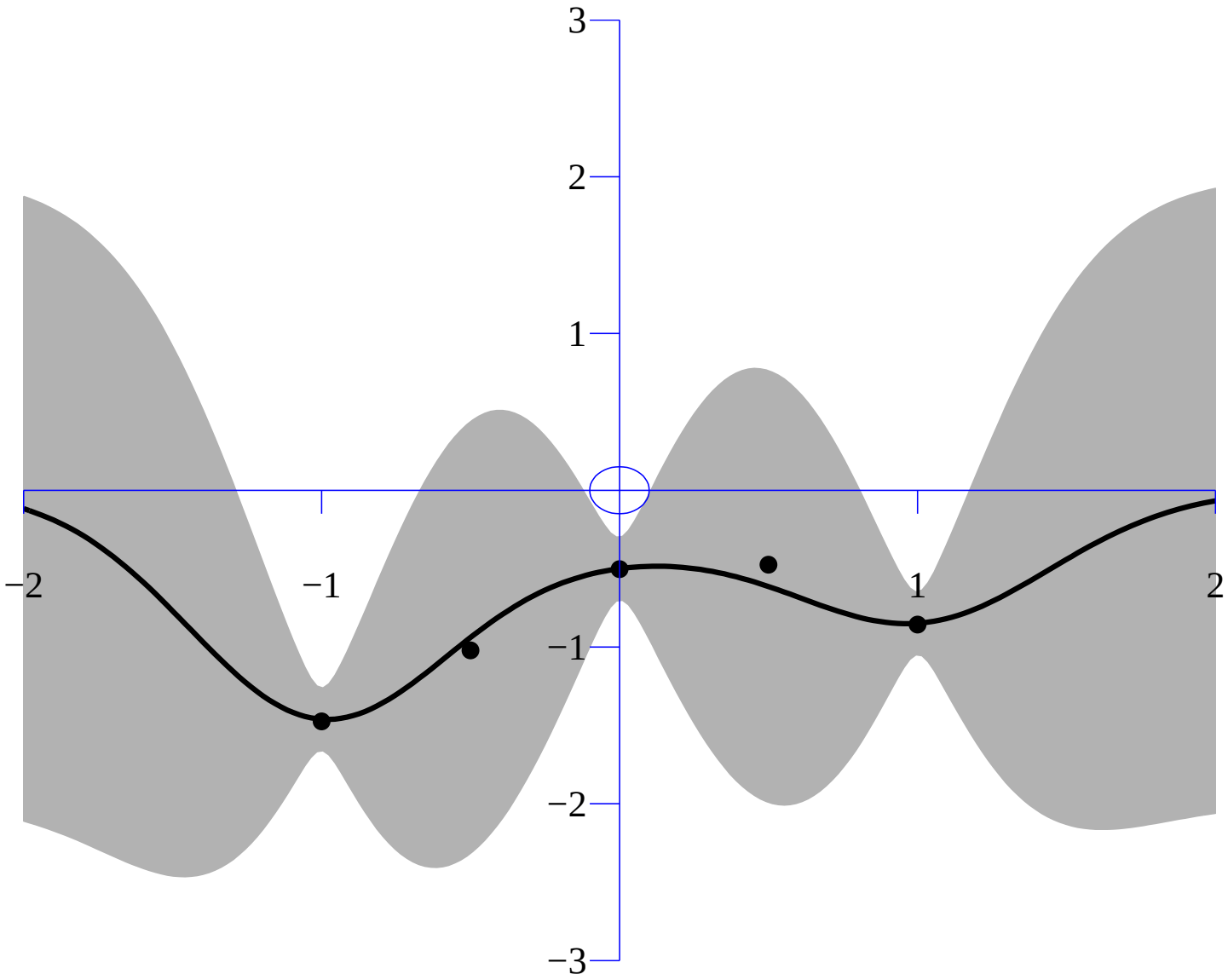
Fitting the data



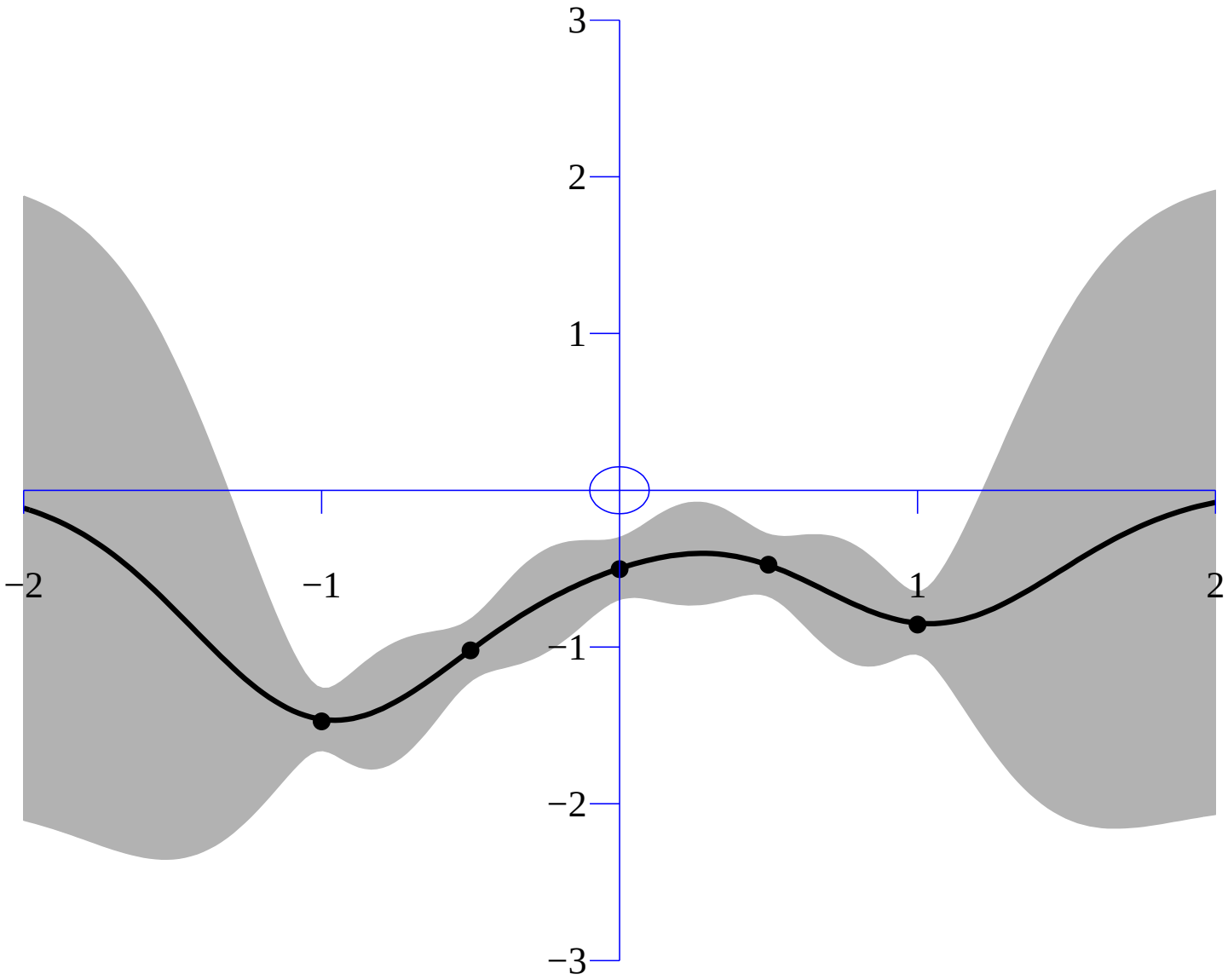
Fitting the data



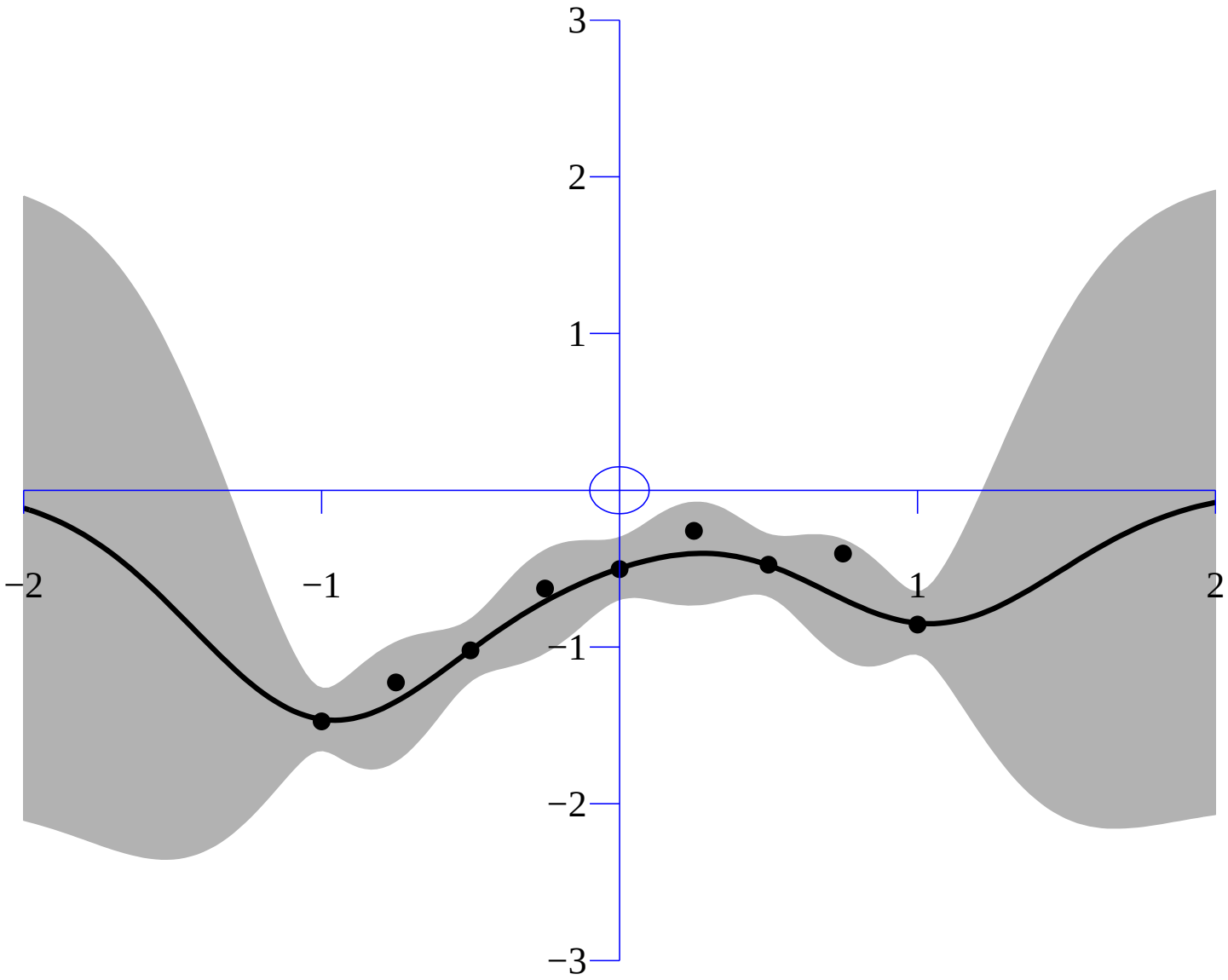
Fitting the data



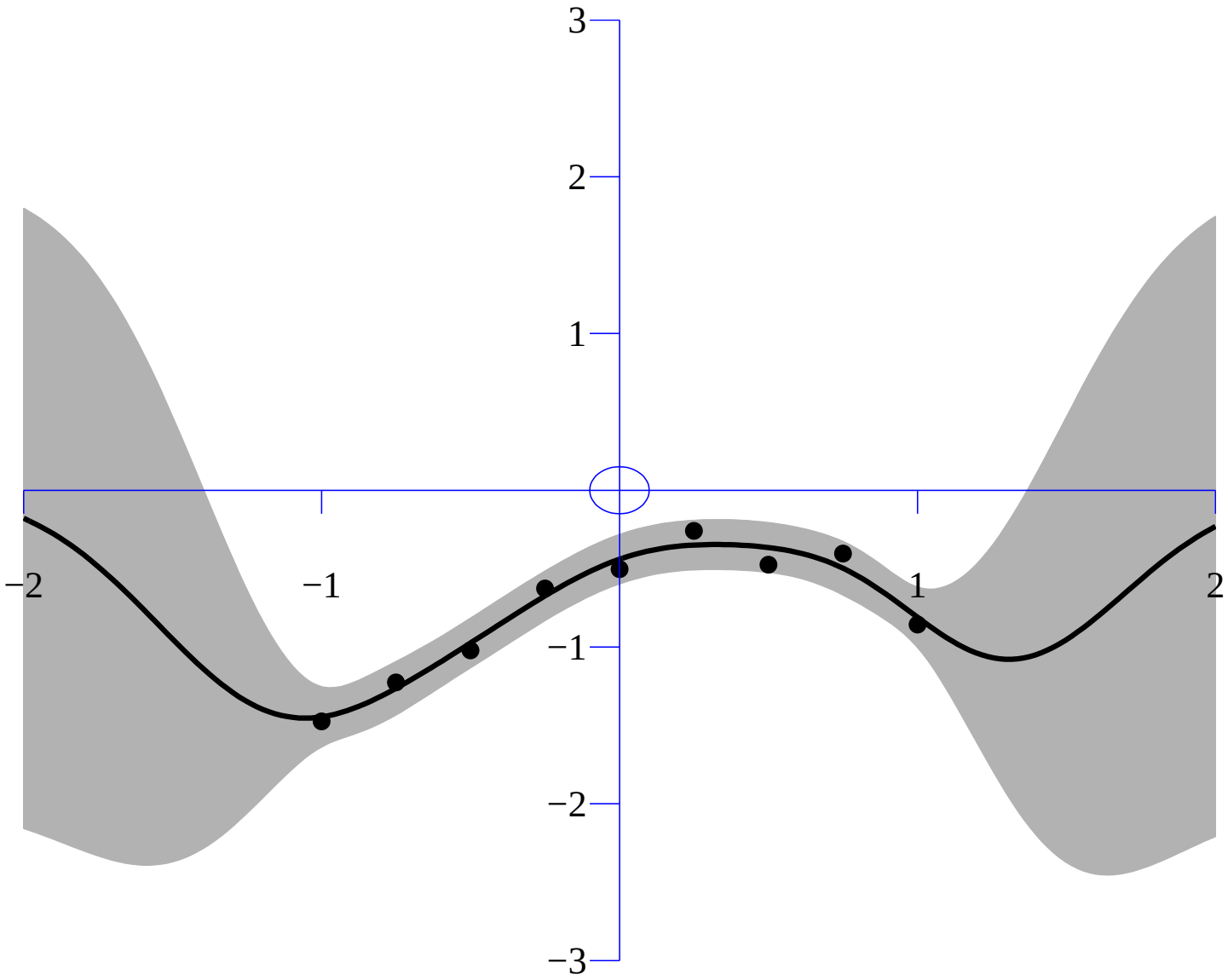
Fitting the data



Fitting the data



Fitting the data



GP learning the kernel

Consider the covariance function K with hyperparameters $\boldsymbol{\theta} = (v_0, v_1, r_1, \ldots, r_d, \alpha)$:

$$K_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) = v_0 \exp\left\{ -\sum_{d=1}^{D} \left(\frac{|x_i^{(d)} - x_j^{(d)}|}{r_d} \right)^{\alpha} \right\} + v_1$$

Given a data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$, how do we learn $\boldsymbol{\theta}$?

The marginal likelihood is a function of $\boldsymbol{\theta}$

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}} + \sigma^2 \mathbf{I})$$

where its log is:

$$\ln p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2} \ln \det(\mathbf{K}_{\boldsymbol{\theta}} + \sigma^2 \mathbf{I}) - \frac{1}{2} \mathbf{y}^\top (\mathbf{K}_{\boldsymbol{\theta}} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} + \text{const}$$

which can be optimized as a function of $\boldsymbol{\theta}$ and σ.

Alternatively, one can infer $\boldsymbol{\theta}$ using Bayesian methods, which is more costly but immune to overfitting.

From linear regression to GPs:

- Linear regression with inputs x_i and outputs y_i: $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}, \qquad y = f(\mathbf{x}) + \epsilon,$
- Linear regression with functions: $y_i = \phi(x_i) + \epsilon_i = \mathcal{N}(\mathbf{y} \mid X\mathbf{w}, \sigma^2\mathbf{I}).$
- Bayesian linear regression with basis functions: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p), \qquad \epsilon_i \sim \mathrm{N}(\cdot|0, \sigma^2)$
- Integrating out the coefficients , we find: $p(\mathbf{y} \mid X) = \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \Phi\Sigma_p\Phi^\top + \sigma^2\mathbf{I}).$
- I f f :

$$p(\mathbf{w} \mid X, \mathbf{y}) \propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - X\mathbf{w})^\top(\mathbf{y} - X\mathbf{w})\right\}\exp\left\{-\frac{1}{2}\mathbf{w}^\top\Sigma_p^{-1}\mathbf{w}\right\} \qquad p(\mathbf{w} \mid X, \mathbf{y}) = \mathcal{N}(\mathbf{w} \mid \frac{1}{\sigma^2}A^{-1}X^\top\mathbf{y}, A^{-1})$$

$$\propto \exp\left\{-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top(\frac{1}{\sigma^2}X^\top X + \Sigma_p^{-1})(\mathbf{w} - \bar{\mathbf{w}})\right\}, \qquad \Rightarrow \qquad A = \sigma^{-2}X^\top X + \Sigma_p^{-1}$$

- I f predictive distribution: $f_* \mid \mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(f_* \mid \frac{1}{\sigma^2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi^\top\mathbf{y}, \phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*)),$

$$\sigma^{-2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi^\top\mathbf{y} = \phi(\mathbf{x}_*)^\top\Sigma_p\Phi^\top(\Phi\Sigma_p\Phi^\top + \sigma^2\mathbf{I})^{-1}\mathbf{y}.$$

$$\phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*) = \phi(\mathbf{x}_*)^\top\Sigma_p\phi(\mathbf{x}_*) - \phi(\mathbf{x}_*)^\top\Sigma_p\Phi^\top(\sigma^2\mathbf{I} + \Phi\Sigma_p\Phi^\top)^{-1}\Phi\Sigma_p\phi(\mathbf{x}_*).$$

From linear regression to GPs:

- Integrating out , : $p(\mathbf{y} \mid X) = \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \Phi\Sigma_p\Phi^\top + \sigma^2\mathbf{I})$.
- I f predictive distribution: $f_* \mid \mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(f_* \mid \frac{1}{\sigma^2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi^\top\mathbf{y}, \phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*))$,

$$\sigma^{-2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi^\top\mathbf{y} = \phi(\mathbf{x}_*)^\top\Sigma_p\Phi^\top(\Phi\Sigma_p\Phi^\top + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*) = \phi(\mathbf{x}_*)^\top\Sigma_p\phi(\mathbf{x}_*) - \phi(\mathbf{x}_*)^\top\Sigma_p\Phi^\top(\sigma^2\mathbf{I} + \Phi\Sigma_p\Phi^\top)^{-1}\Phi\Sigma_p\phi(\mathbf{x}_*).$$

- $\varphi(\mathbf{x}) = \Sigma_p^{1/2}\phi(\mathbf{x})$ $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top\Sigma_p\phi(\mathbf{x})$
- Gaussian process $p(\mathbf{y} \mid X) = \mathcal{N}(\mathbf{0}, K(X, X) + \sigma^2\mathbf{I})$

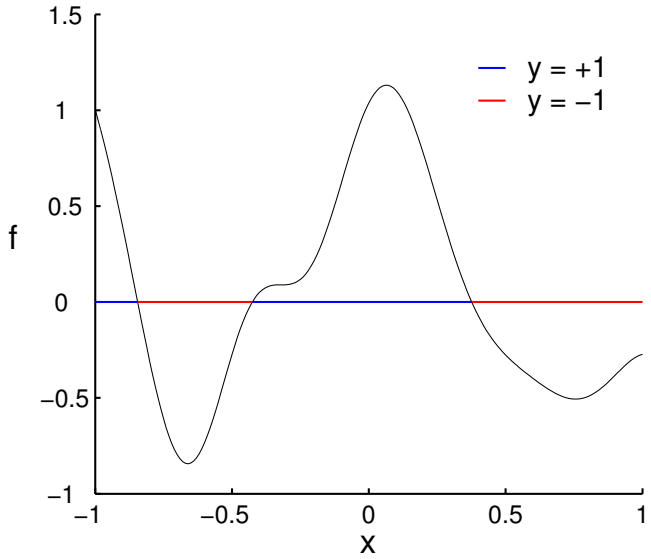
$$p(f_* \mid \mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}\Big(\mathbf{k}(\mathbf{x}_*, X)\big[K(X, X) + \sigma^2\mathbf{I}\big]^{-1}\mathbf{y},$$
$$k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*, X)\big[K(X, X) + \sigma^2\mathbf{I}\big]^{-1}\mathbf{k}(X, \mathbf{x}_*)\Big)$$

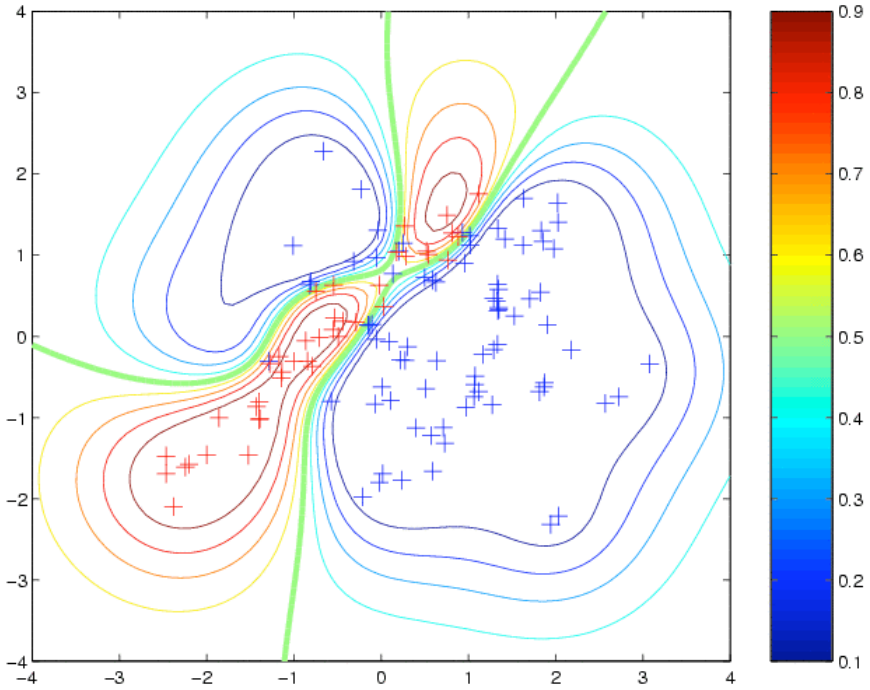
$$p(\mathbf{f}_* \mid X_*, X, \mathbf{y}) = \mathcal{N}\Big(K(X_*, X)\big[K(X, X) + \sigma^2\mathbf{I}\big]^{-1}\mathbf{y},$$
$$K(X_*, X_*) - K(X_*, X)\big[K(X, X) + \sigma^2\mathbf{I}\big]^{-1}K(X, X_*)\Big).$$

A multilayer perceptron (neural network) with infinitely many hidden units and Gaussian priors on the weights → a GP (Neal, 1996)

Using Gaussian Processes for Classification

Binary classification problem: Given a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with binary class labels $y_i \in \{-1, +1\}$, infer class label probabilities at new points.





There are many ways to relate function values $f_i = f(\mathbf{x}_i)$ to class probabilities:

$$p(y_i|f_i) = \begin{cases} \frac{1}{1+\exp(-y_i f_i)} & \text{sigmoid (logistic)} \\ \Phi(y_i f_i) & \text{cumulative normal} \end{cases}$$

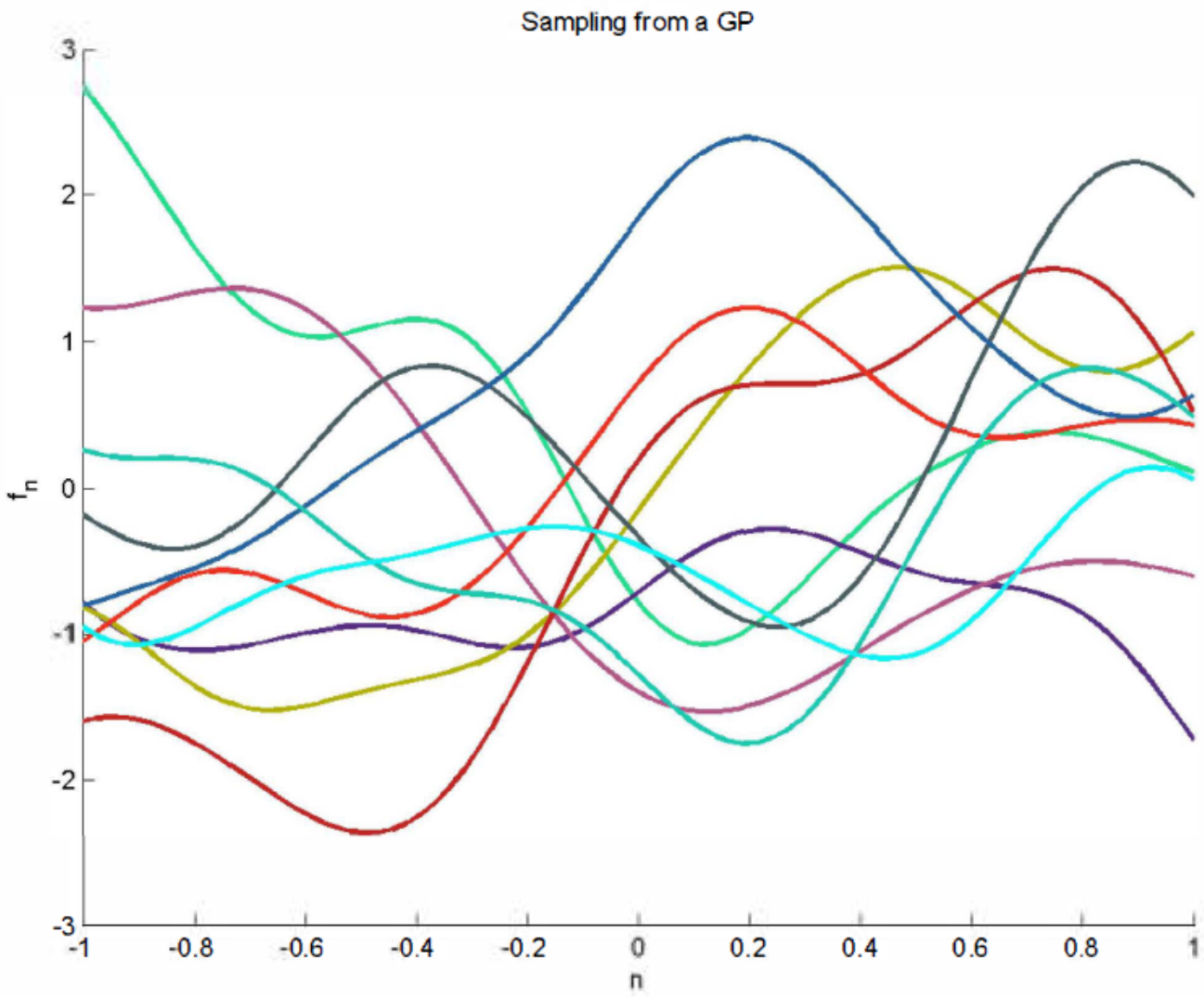
Non-Gaussian likelihood, so we need to use approximate inference methods (Laplace, EP, MCMC).

Conclusions

- Gaussian processes define distributions on functions which can be used for nonlinear regression, classification, ranking, preference learning, ordinal regression, etc.
- GPs are closely related to many other models. We can derive them from:
  - Bayesian kernel machines
  - Linear regression with basis functions
  - Infinite multi-layer perceptron neural networks
- Compared to SVMs, GPs offer several advantages: learning the kernel and regularization parameters, integrated feature selection, fully probabilistic predictions, interpretability.

Practical applications of supervised learning frequently involve situations exhibiting an order among the different categories, e.g. a teacher always rates his/her students by giving grades on their overall performance. In contrast to metric regression problems, the grades are usually discrete and finite. These grades are also different from the class labels in classification problems due to the existence of ranking information. For example, grade labels have the ordering $F < D < C < B < A$. This is a learning task of predicting variables of ordinal scale, a setting bridging between metric regression and classification referred to as *ranking learning* or *ordinal regression*.

Homework： Draw samples from a defined GP like this! Give the details of the GP settings, such as kernel function, mean function and hyper-parameters.



References

- Qi, Y., Minka, T.P., Picard, R.W., and Ghahramani, Z. (2004) Predictive Automatic Relevance Determination by Expectation Propagation. In **Twenty-first International Conference on Machine Learning** (ICML-04). Banff, Alberta, Canada.
- Quiñonero-Candela, J. and Rasmussen, C.E. (2005) A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* **6**:1959.
- Naish-Guzman, A. and Holden, S. (2008) The generalized FITC approximation. Advances in Neural Information Processing Systems 20:1057–1064.
- Neal, R. M. (1996) Bayesian learning for neural networks. Springer Verlag.
- Neal, R. M. (1998). Regression and classification using Gaussian process priors (with discussion). In Bernardo, J. M. et al., editors, **Bayesian statistics 6**, pages 475-501. Oxford University Press.
- O'Hagan, A. (1978). Curve Fitting and Optimal Design for Prediction (with discussion). **Journal of the Royal Statistical Society B**, 40(1):1-42.
- Rasmussen, C.E. and Williams, C.K.I. (2006) Gaussian Processes for Machine Learning. MIT Press.
- Snelson, E. and Ghahramani, Z. (2006a) Sparse Gaussian Processes using Pseudo-Inputs. In **Advances in Neural Information Processing Systems 18** (NIPS-2005).
- Snelson, E. and Ghahramani, Z. (2006b) Variable noise and dimensionality reduction for sparse Gaussian processes. In **Uncertainty in Artifical Intelligence 22** (UAI).
- More information and code at: `http://www.gaussianprocess.org/`