



7. Gaussian Processes - Regression

Definition

Definition: A **Gaussian process** is a collection of random variables, any finite number of which have a joint Gaussian distribution.

The number of random variables can be **infinite!**

This means: a GP is a Gaussian distribution over **functions!**

To specify a GP we need:

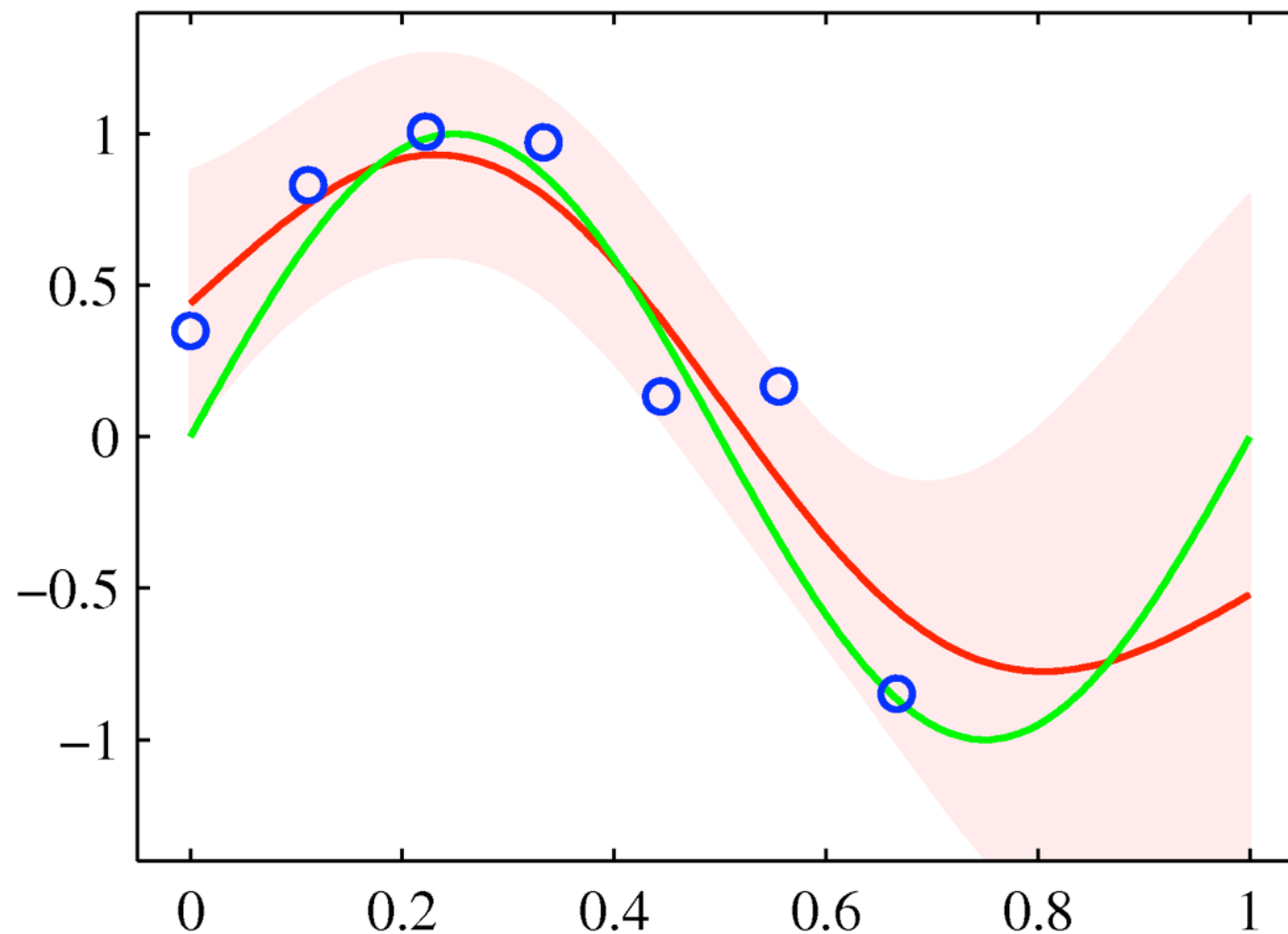
mean function: $m(\mathbf{x}) = \mathbb{E}[y(\mathbf{x})]$

covariance function:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \mathbb{E}[y(\mathbf{x}_1) - m(\mathbf{x}_1)y(\mathbf{x}_2) - m(\mathbf{x}_2)]$$



Example



- green line: sinusoidal data source
- blue circles: data points with Gaussian noise
- red line: mean function of the Gaussian process
- shaded red area: 2σ confidence interval



How Can We Handle Infinity?

Idea: split the (infinite) number of random variables into a finite and an infinite subset.

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_f \\ \mathbf{x}_i \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \boldsymbol{\mu}_f \\ \boldsymbol{\mu}_i \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_f & \boldsymbol{\Sigma}_{fi} \\ \boldsymbol{\Sigma}_{fi}^T & \boldsymbol{\Sigma}_i \end{pmatrix} \right)$$

finite part

infinite part

From the **marginalization property** we get:

$$p(\mathbf{x}_f) = \int p(\mathbf{x}_f, \mathbf{x}_i) d\mathbf{x}_i = \mathcal{N}(\mathbf{x}_f \mid \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$$

This means we can use finite vectors.



The Covariance Function

The most used covariance function (kernel) is:

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2l^2} (\mathbf{x}_p - \mathbf{x}_q)^2\right) + \sigma_n^2 \delta_{pq}$$

signal variance

length scale

noise variance

It is known as “squared exponential”, “radial basis function” or “Gaussian kernel”.

Other possibilities exist, e.g. the exponential kernel:

$$k(\mathbf{x}_p, \mathbf{x}_q) = \exp(-\theta |\mathbf{x}_p - \mathbf{x}_q|)$$

This is used in the “Ornstein-Uhlenbeck” process.



Sampling from a GP

Just as we can sample from a Gaussian distribution, we can also generate samples from a GP. **Every sample will then be a function!**

Process:

1. Choose a number of input points $\mathbf{x}_1^*, \dots, \mathbf{x}_M^*$

2. Compute the covariance matrix K where

$$K_{ij} = k(\mathbf{x}_i^*, \mathbf{x}_j^*)$$

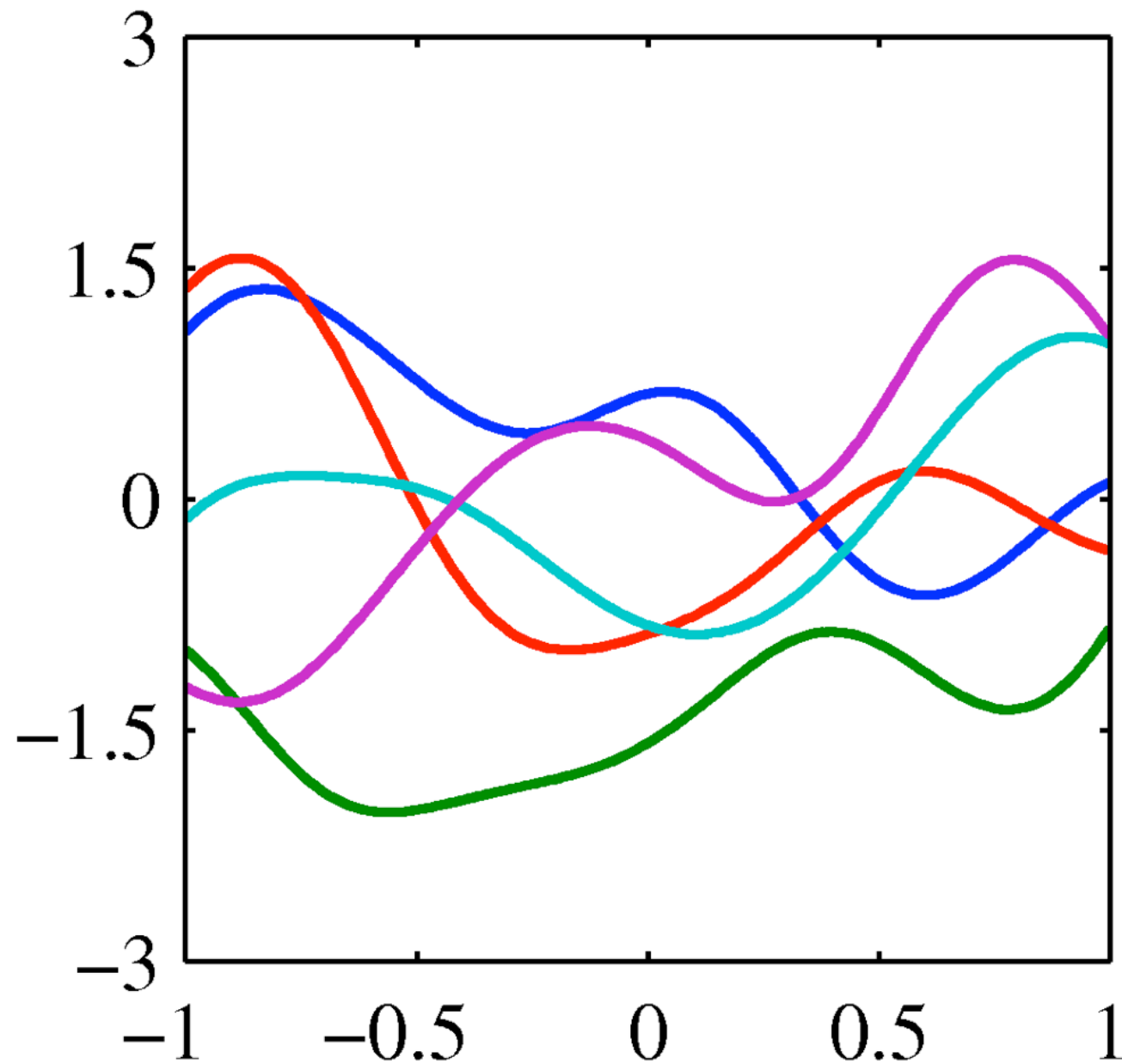
3. Generate a random Gaussian vector from

$$\mathbf{y}_* \sim \mathcal{N}(\mathbf{0}, K)$$

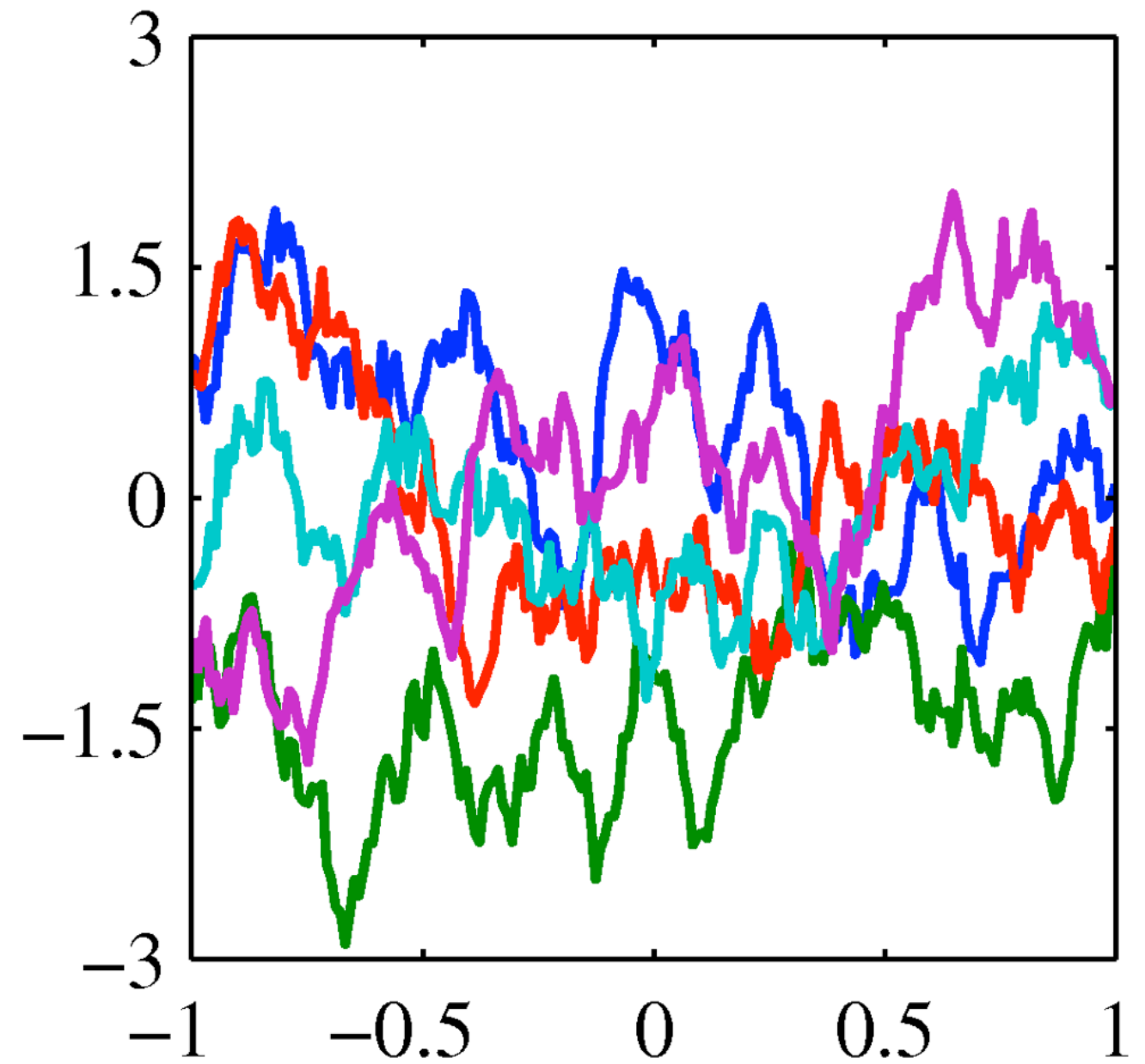
4. Plot the values $\mathbf{x}_1^*, \dots, \mathbf{x}_M^*$ versus y_1^*, \dots, y_M^*



Sampling from a GP



Squared exponential kernel



Exponential kernel



Prediction with a Gaussian Process

Most often we are more interested in predicting new function values for given input data.

We have:

training data $\mathbf{x}_1, \dots, \mathbf{x}_N$ t_1, \dots, t_N

test input $\mathbf{x}_1^*, \dots, \mathbf{x}_M^*$

And we want test outputs y_1^*, \dots, y_M^*

The joint probability is

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{y}_* \end{pmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{pmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{pmatrix} \right)$$

and we need to compute $p(\mathbf{y}^* | \mathbf{x}^*, X, \mathbf{y})$.



Prediction with a Gaussian Process

In the case of only one test point \mathbf{x}^* we have

$$K(X, \mathbf{x}^*) = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_*) \\ \vdots \\ k(\mathbf{x}_N, \mathbf{x}_*) \end{pmatrix} = \mathbf{k}_*$$

Now we compute the conditional distribution

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

where

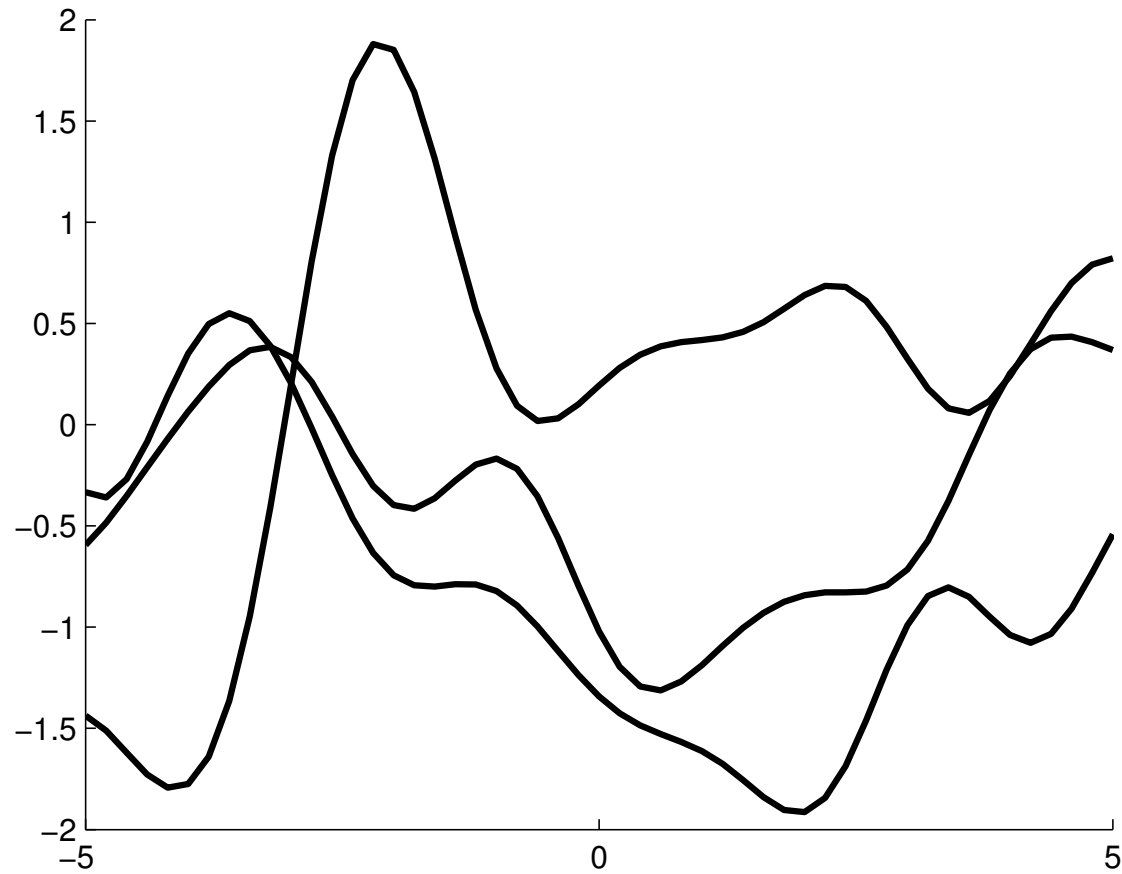
$$\mu_* = \mathbf{k}_*^T K^{-1} \mathbf{t}$$

$$\Sigma_* = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T K^{-1} \mathbf{k}_*$$

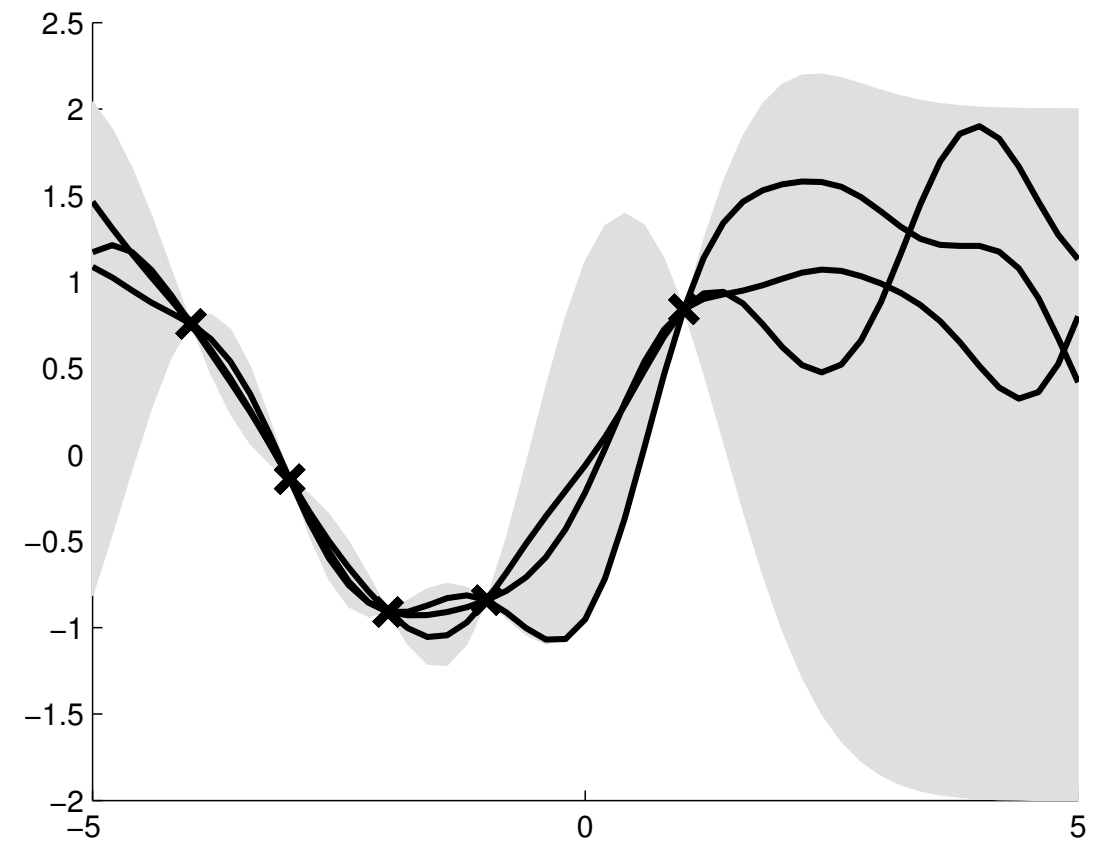
This defines the **predictive distribution**.



Example



Functions sampled from a Gaussian Process prior

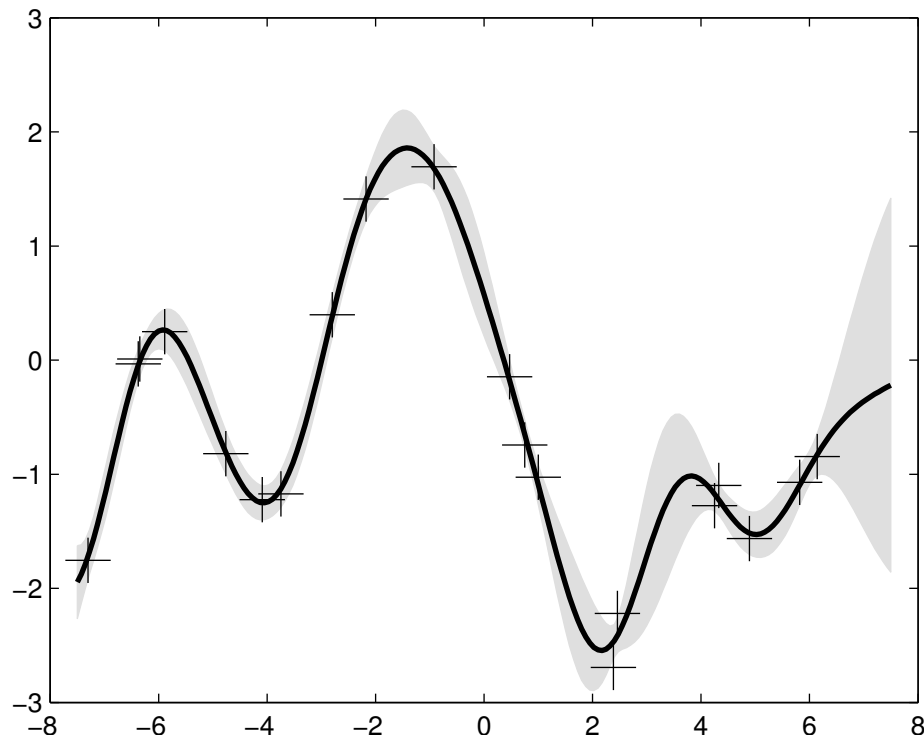


Functions sampled from the predictive distribution

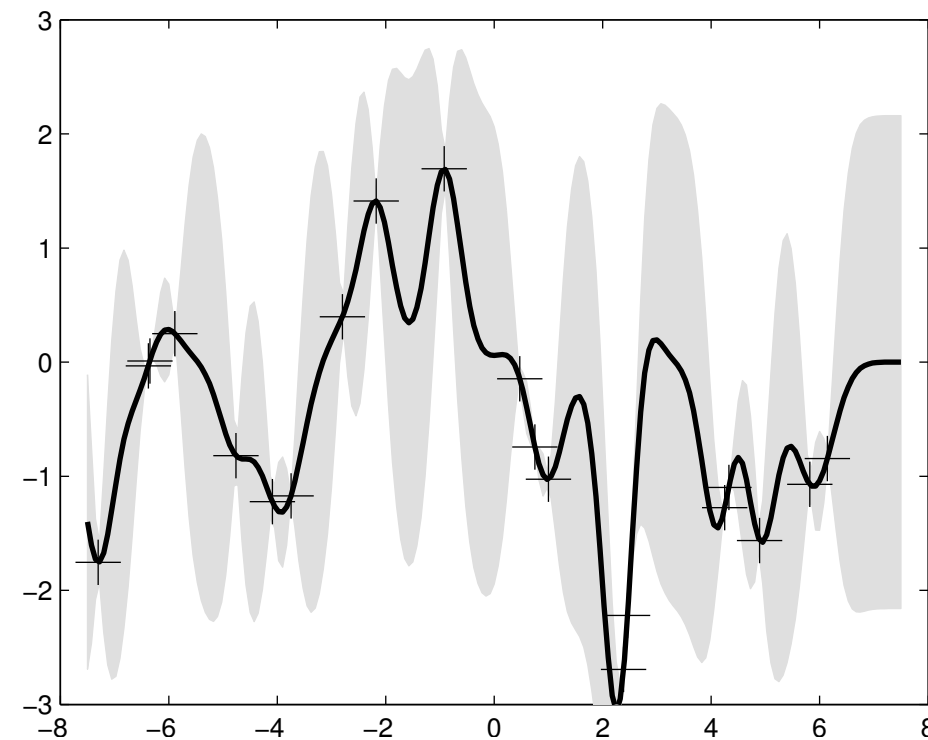
The predictive distribution is itself a Gaussian process. It represents the posterior after observing the data. The covariance is low in the vicinity of data points.



Varying the Hyperparameters

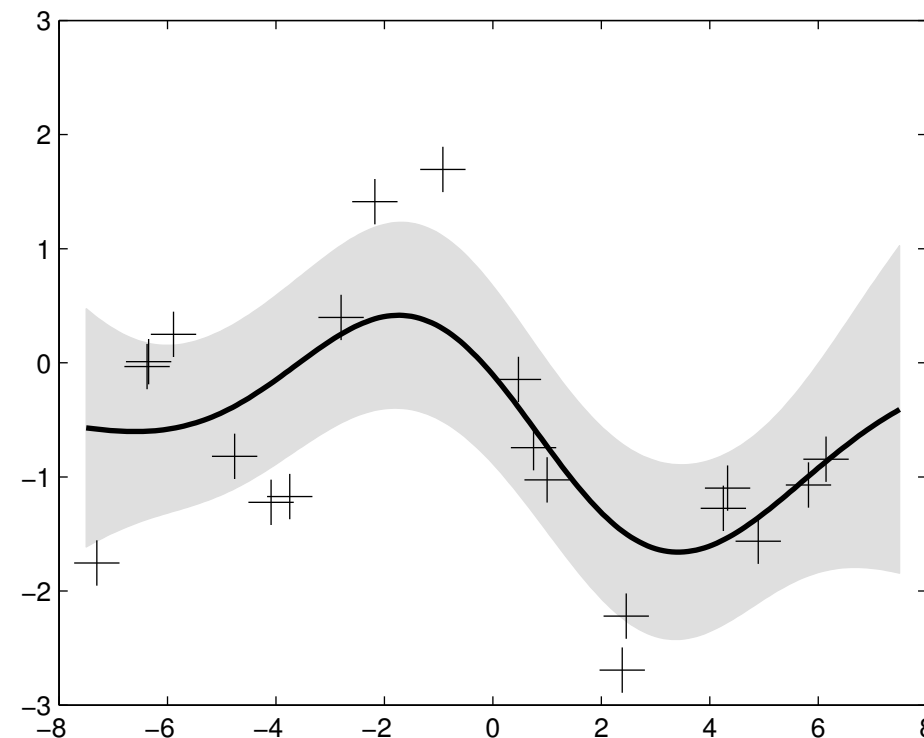


$$l = \sigma_f = 1, \quad \sigma_n = 0.1$$



$$l = 0.3,$$
$$\sigma_f = 1.08,$$
$$\sigma_n = 0.0005$$

- 20 data samples
- GP prediction with different kernel hyper parameters



$$l = 3$$
$$\sigma_f = 1.16$$
$$\sigma_n = 0.89$$





Gaussian Processes - Classification

Gaussian Processes For Classification

In regression we have $y \in \mathbb{R}$, in binary classification we have $y \in \{-1; 1\}$

To use a GP for classification, we can apply a **sigmoid** function to the posterior obtained from the GP and compute the class probability as:

$$p(y = +1 | \mathbf{x}) = \sigma(f(\mathbf{x}))$$

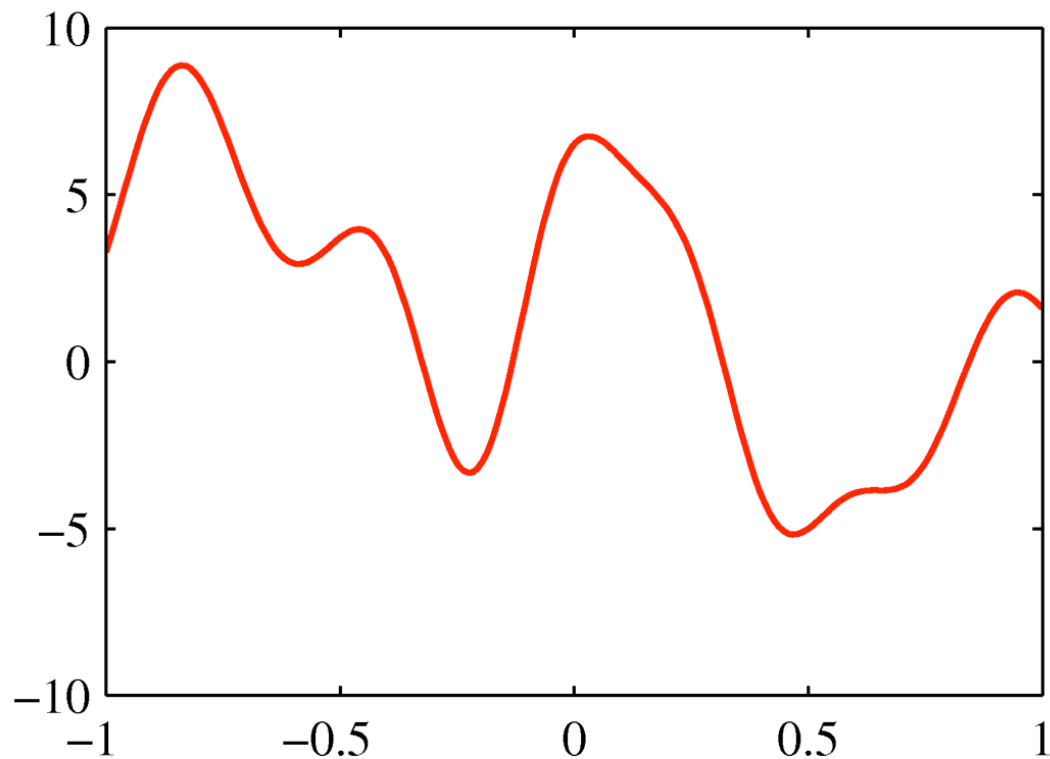
If the sigmoid function is symmetric: $\sigma(-z) = 1 - \sigma(z)$ then we have $p(y | \mathbf{x}) = \sigma(yf(\mathbf{x}))$.

A typical type of sigmoid function is the logistic sigmoid:

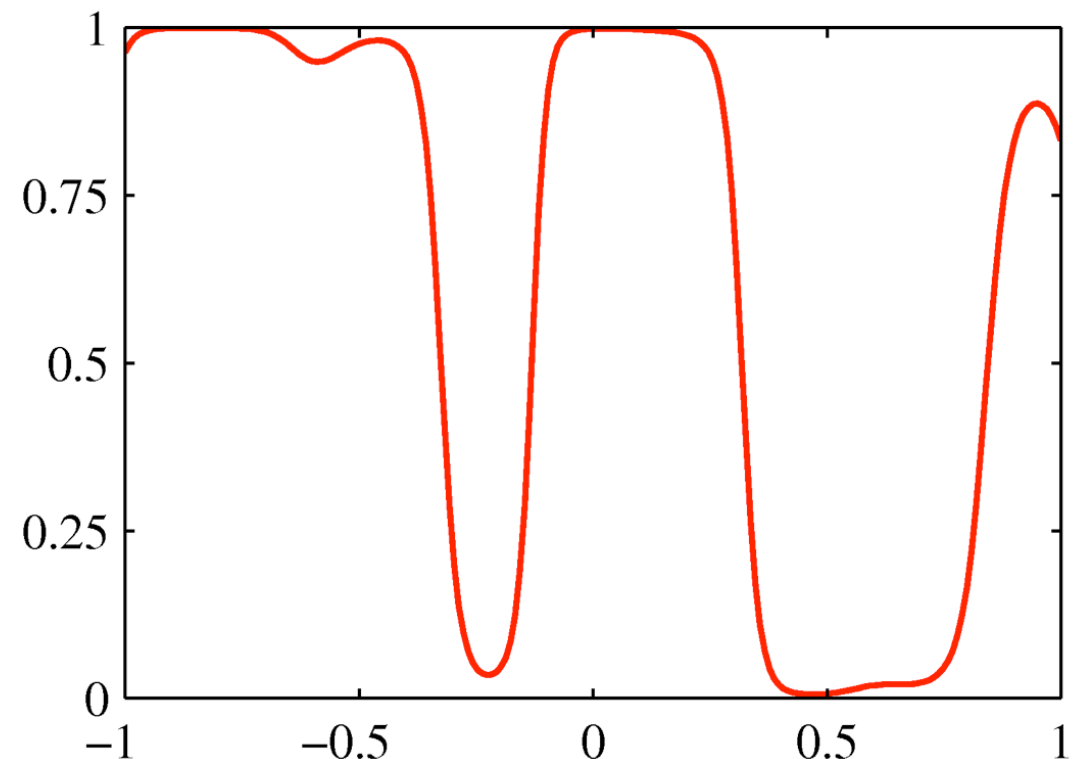
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



Application of the Sigmoid Function



Function sampled from
a Gaussian Process



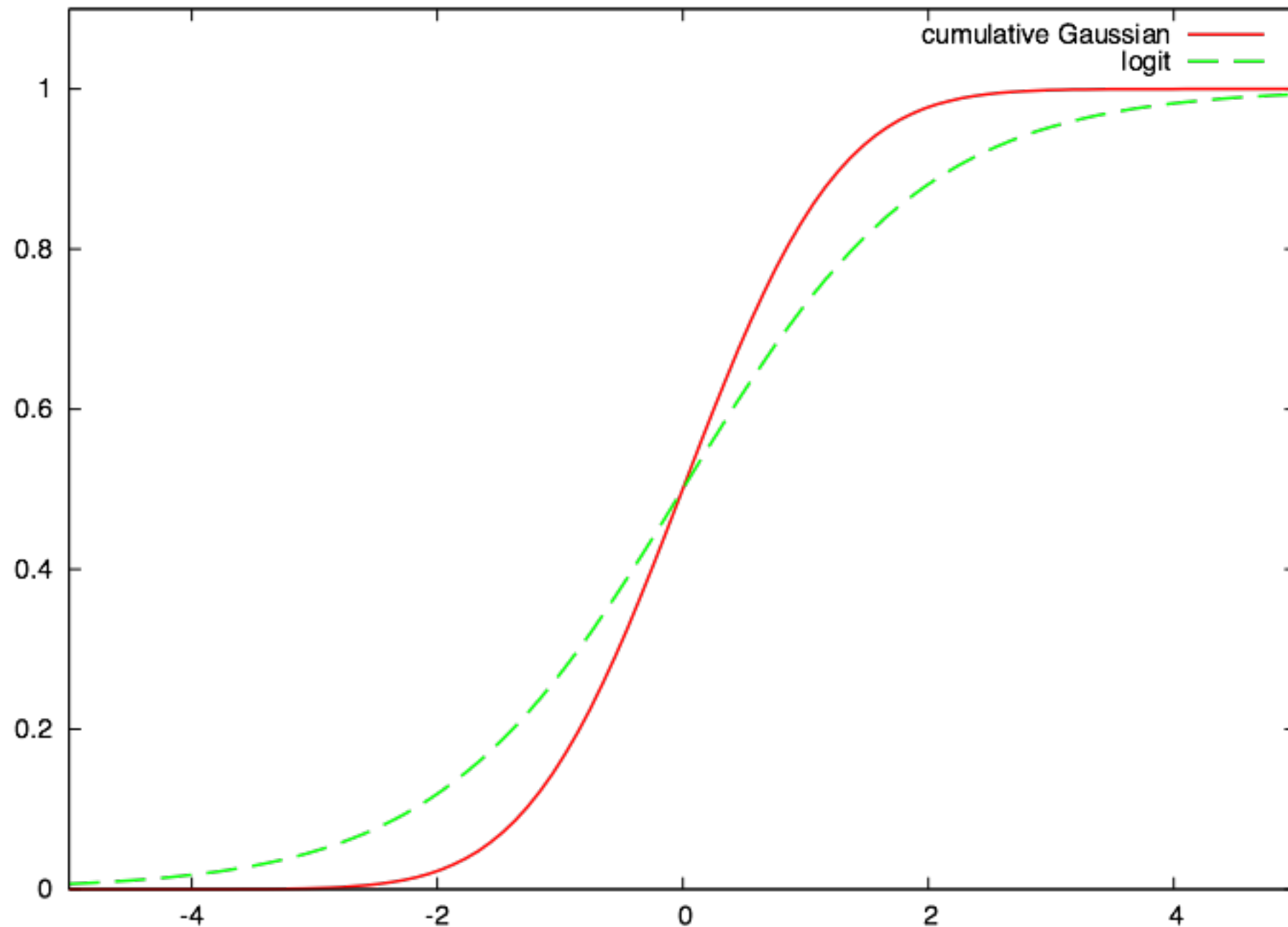
Sigmoid function applied to
the GP function

Another symmetric sigmoid function is the **cumulative Gaussian**:

$$\Phi(z) = \int_{-\infty}^z \mathcal{N}(x \mid 0, 1) dx$$



Visualization of Sigmoid Functions



The cumulative Gaussian is slightly steeper than the logistic sigmoid



The Latent Variables

In regression, we directly estimated f as

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

and values of f where observed in the training data. Now only labels +1 or -1 are observed and f is treated as a set of **latent variables**.

A major advantage of the Gaussian process classifier over other methods is that it **marginalizes** over all latent functions rather than maximizing some model parameters.



Class Prediction with a GP

The aim is to compute the predictive distribution

$$p(y_* = +1 \mid X, \mathbf{y}, \mathbf{x}_*) = \int p(y_* \mid f_*) p(f_* \mid X, \mathbf{y}, \mathbf{x}_*) df_*$$


$$\sigma(f_*)$$



Class Prediction with a GP

The aim is to compute the predictive distribution

$$p(y_* = +1 \mid X, \mathbf{y}, \mathbf{x}_*) = \int p(y_* \mid f_*)p(f_* \mid X, \mathbf{y}, \mathbf{x}_*)df_*$$

we marginalize over the latent variables from the training data:

$$p(f_* \mid X, \mathbf{y}, \mathbf{x}_*) = \int p(f_* \mid X, \mathbf{x}_*, \mathbf{f})p(\mathbf{f} \mid X, \mathbf{y})d\mathbf{f}$$

predictive distribution of the latent variable (from regression)



Class Prediction with a GP

The aim is to compute the predictive distribution

$$p(y_* = +1 \mid X, \mathbf{y}, \mathbf{x}_*) = \int p(y_* \mid f_*) p(f_* \mid X, \mathbf{y}, \mathbf{x}_*) df_*$$

we marginalize over the latent variables from the training data:

$$p(f_* \mid X, \mathbf{y}, \mathbf{x}_*) = \int p(f_* \mid X, \mathbf{x}_*, \mathbf{f}) p(\mathbf{f} \mid X, \mathbf{y}) d\mathbf{f}$$

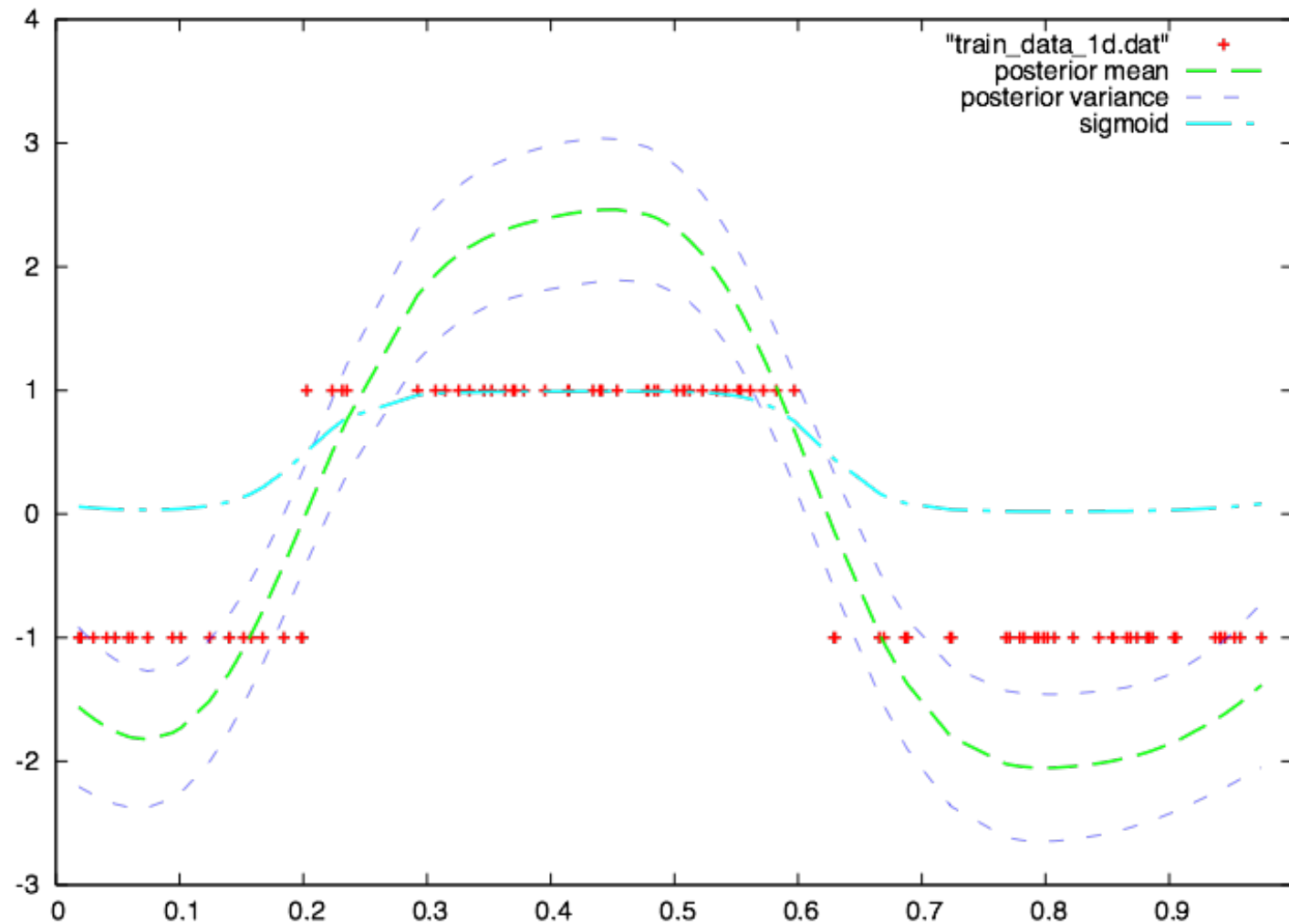
we need the posterior over the latent variables:

$$p(\mathbf{f} \mid X, \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X)}{p(\mathbf{y} \mid X)}$$

likelihood (sigmoid) → prior → normalizer



A Simple Example



- Red: Two-class training data
- Green: mean function of $p(\mathbf{f} \mid X, \mathbf{y})$
- Light blue: sigmoid of the mean function



But There Is A Problem...

$$p(\mathbf{f} \mid X, \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid X)}{p(\mathbf{y} \mid X)}$$

- The likelihood term is not a Gaussian!
- This means, we can not compute the posterior in closed form.
- There are several different solutions in the literature, e.g.:
 - Laplace approximation
 - Expectation Propagation
 - Variational methods



Predictions

Now that we have $p(\mathbf{f} | X, \mathbf{y})$ we can compute:

$$p(f_* | X, \mathbf{y}, \mathbf{x}_*) = \int p(f_* | X, \mathbf{x}_*, \mathbf{f}) p(\mathbf{f} | X, \mathbf{y}) d\mathbf{f}$$

From the regression case we have:

$$p(f_* | X, \mathbf{x}_*, \mathbf{f}) = \mathcal{N}(f_* | \mu_*, \Sigma_*)$$

where $\mu_* = \mathbf{k}_*^T K^{-1} \mathbf{f}$ $\Sigma_* = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T K^{-1} \mathbf{k}_*$

Linear in \mathbf{f}

This means we can use a particular property of Gaussians:



A Convenient Gaussian Property

If we are given this:

I. $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mu, \Sigma_1)$

II. $p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid A\mathbf{x} + \mathbf{b}, \Sigma_2)$

Then it follows (Bayes Rule for Gaussians):

III. $p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid A\mu + \mathbf{b}, \Sigma_2 + A\Sigma_1A^T)$

IV. $p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}(\mathbf{x} \mid \Sigma(A^T \Sigma_2^{-1}(\mathbf{y} - \mathbf{b}) + \Sigma_1^{-1}\mathbf{y}), \Sigma)$

where

$$\Sigma = (\Sigma_1^{-1} + A^T \Sigma_2^{-1} A)^{-1}$$



Applying this to Laplace

$$\mathbb{E}[f_* \mid X, \mathbf{y}, \mathbf{x}_*] = \mathbf{k}(\mathbf{x}_*)^T K^{-1} \hat{\mathbf{f}}$$

$$\mathbb{V}[f_* \mid X, \mathbf{y}, \mathbf{x}_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + W^{-1})^{-1} \mathbf{k}_*$$

It remains to compute

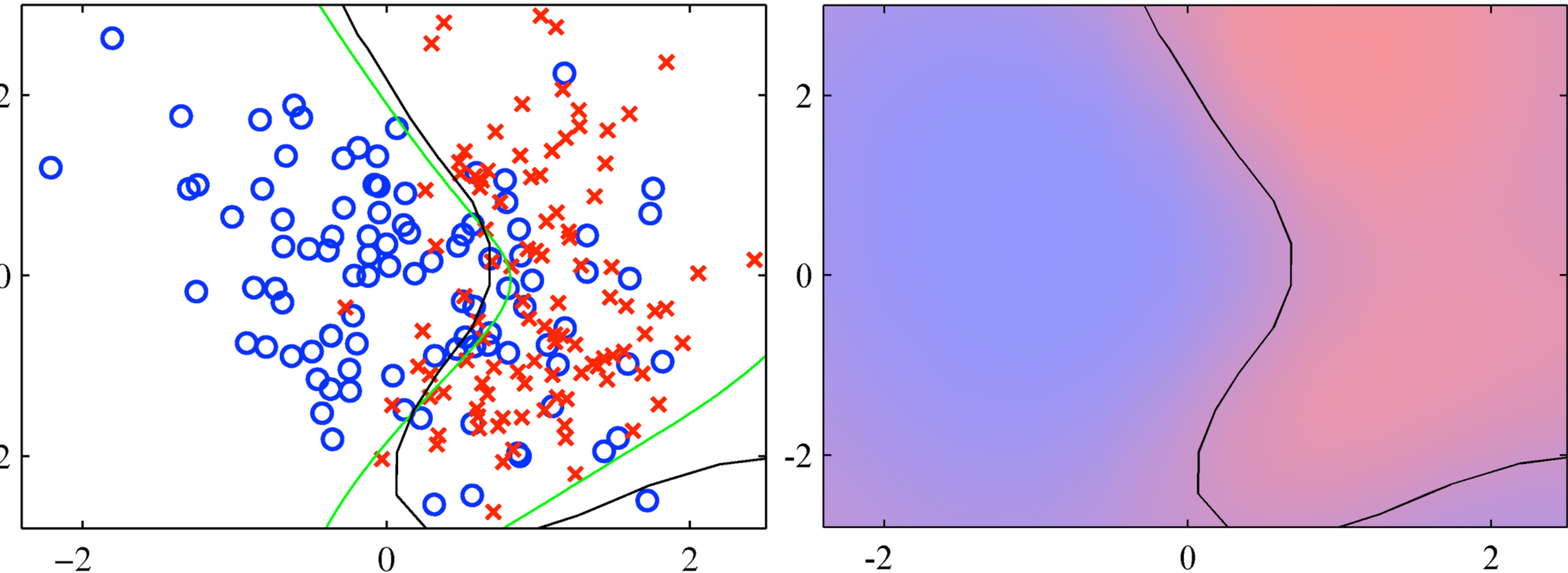
$$p(y_* = +1 \mid X, \mathbf{y}, \mathbf{x}_*) = \int p(y_* \mid f_*) p(f_* \mid X, \mathbf{y}, \mathbf{x}_*) df_*$$

Depending on the kind of sigmoid function we

- can compute this in closed form (cumulative Gaussian sigmoid)
- have to use sampling methods or analytical approximations (logistic sigmoid)



A Simple Example



- Two-class problem (training data in red and blue)
- Green line: optimal decision boundary
- Black line: GP classifier decision boundary
- Right: posterior probability



Summary

- Gaussian Processes are Normal distributions over functions
- To specify a GP we need a covariance function (kernel) and a mean function
- For regression we can compute the predictive distribution in closed form
- For classification, we use a sigmoid and have to approximate the latent posterior
- More on Gaussian Processes:
http://videlectures.net/epsrws08_rasmussen_lgp/

