

# Gaussian Processes

Thibault Randrianarisoa

University of Toronto, Winter 2026

March 10, 2026



# Overview

First, we recall the basics of linear regression.

Then, we discuss kernel functions.

We end up with learning for Gaussian processes.

# Reminder: Linear Regression

---

# Completing the Square for Gaussians

Useful technique to find moments of Gaussian random variables.

- It is a multivariate generalization of completing the square.
- The density of  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  satisfies:

$$\begin{aligned}\log p(\mathbf{x}) &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) + \text{const} \\ &= -\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \text{const}\end{aligned}$$

- Thus, if we know  $\mathbf{w}$  is Gaussian with *unknown* mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ , and we also know that

$$\log p(\mathbf{w}) = -\frac{1}{2}\mathbf{w}^\top \mathbf{A}\mathbf{w} + \mathbf{w}^\top \mathbf{b} + \text{const},$$

then  $\boldsymbol{\Sigma} = \mathbf{A}^{-1}$ ,  $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \mathbf{b}$  and so

$$\mathbf{w} \sim \mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}).$$

# Bayesian Linear Regression

- We take the Bayesian approach to linear regression.
  - This is in contrast with the standard regression.
  - By inferring a posterior distribution over the *parameters*, the model can know what it doesn't know.
- How can uncertainty in the predictions help us?
  - Smooth out the predictions by averaging over lots of plausible explanations
  - Assign confidences to predictions
  - Make more robust decisions

## Reminder: Linear Regression

- Given a training set of inputs and targets  $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$
- Linear model:

$$y = \mathbf{w}^\top \psi(\mathbf{x}) + \epsilon$$

where  $\psi(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^M$  is the feature map,  $\mathbf{w} \in \mathbb{R}^M$ .

- We have the design matrix  $\mathbf{X} \in \mathbb{R}^{N \times D}$  in input space and the feature matrix and outputs

$$\Psi = \begin{bmatrix} \cdots & \psi(\mathbf{x}^{(1)}) & \cdots \\ \cdots & \psi(\mathbf{x}^{(2)}) & \cdots \\ & \vdots & \\ \cdots & \psi(\mathbf{x}^{(N)}) & \cdots \end{bmatrix} \in \mathbb{R}^{N \times M}, \quad \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}.$$

Predictions are

$$\hat{\mathbf{y}} = \Psi \mathbf{w}.$$

## Reminder: Ridge Regression

- Penalized sum of squares (ridge regression),  $\lambda \geq 0$ :

$$\text{minimize } \frac{1}{2} \|\mathbf{y} - \Psi \mathbf{w}\|^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- The gradient:  $(\Psi^\top \Psi + \lambda \mathbf{I}) \mathbf{w} - \Psi^\top \mathbf{y}$ .
- Solution 1: solve analytically by setting the gradient to 0

$$\mathbf{w} = (\Psi^\top \Psi + \lambda \mathbf{I})^{-1} \Psi^\top \mathbf{y}$$

$\in \mathbb{R}^{M \times M}$

- Solution 2: solve approximately using gradient descent

$$\mathbf{w} \leftarrow (1 - \alpha \lambda) \mathbf{w} - \alpha \Psi^\top (\Psi \mathbf{w} - \mathbf{y})$$

**deterministic** → **probabilistic** → **Bayesian**

We first recall the standard probabilistic reformulation of this model. Then make this Bayesian.

# Linear Regression as Maximum Likelihood

- We can give linear regression a probabilistic interpretation by assuming a Gaussian noise model:

$$y \mid \mathbf{x} \sim \mathcal{N}(\mathbf{w}^\top \psi(\mathbf{x}), \sigma^2)$$

- Linear regression is just maximum log-likelihood under this model:

$$\begin{aligned} \sum_{i=1}^N \log p(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w}, b) &= \sum_{i=1}^N \log \mathcal{N}(y^{(i)}; \mathbf{w}^\top \psi(\mathbf{x}^{(i)}), \sigma^2) \\ &= \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{(y^{(i)} - \mathbf{w}^\top \psi(\mathbf{x}^{(i)}))^2}{2\sigma^2} \right) \right] \\ &= \text{const} - \frac{1}{2\sigma^2} \sum_{i=1}^N (y^{(i)} - \mathbf{w}^\top \psi(\mathbf{x}^{(i)}))^2 \\ &= \text{const} - \frac{1}{2\sigma^2} \|\mathbf{y} - \Psi \mathbf{w}\|^2 \end{aligned}$$

$\Psi \mathbf{w}$

# Regularized Linear Regression as MAP Estimation

maximum a posteriori

- View an  $L_2$  regularizer as MAP inference with a Gaussian prior ( $p(\mathbf{w} | \mathcal{D}) \propto p(\mathbf{w})p(\mathcal{D} | \mathbf{w})$ ).

$$\arg \max_{\mathbf{w}} \log p(\mathbf{w} | \mathcal{D}) = \arg \max_{\mathbf{w}} [\log p(\mathbf{w}) + \log p(\mathcal{D} | \mathbf{w})]$$

prior

- We just derived the likelihood term  $\log p(\mathcal{D} | \mathbf{w})$ :

$$\log p(\mathcal{D} | \mathbf{w}) = \text{const} - \frac{1}{2\sigma^2} \|\mathbf{y} - \Psi\mathbf{w}\|^2$$

- Assume a Gaussian prior,  $\mathbf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$ :

$$\begin{aligned} \log p(\mathbf{w}) &= \log \left[ \frac{1}{(2\pi)^{D/2} |\mathbf{S}|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{w} - \mathbf{m}) \right) \right] \\ &= -\frac{1}{2} (\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{w} - \mathbf{m}) + \text{const} \end{aligned}$$

- Commonly,  $\mathbf{m} = \mathbf{0}$  and  $\mathbf{S} = \eta \mathbf{I}$ , so

$$\log p(\mathbf{w}) = -\frac{1}{2\eta} \|\mathbf{w}\|^2 + \text{const.}$$

Ridge regression

minimize

$$\frac{1}{2\sigma^2} \|\mathbf{y} - \Psi\mathbf{w}\|^2 + \frac{1}{2\eta} \|\mathbf{w}\|^2$$

This is just  $L_2$  regularization!

# Full Bayesian Inference

- Full Bayesian inference makes predictions by averaging over all likely explanations under the posterior distribution.

- Compute posterior using Bayes' Rule:  $p(\mathbf{w} | \mathcal{D}) \propto p(\mathbf{w})p(\mathcal{D} | \mathbf{w})$

- Make predictions using the **posterior predictive distribution**:

$$p(y | \mathbf{x}, \mathcal{D}) = \int p(\mathbf{w} | \mathcal{D}) p(y | \mathbf{x}, \mathbf{w}) d\mathbf{w}$$

- Doing this lets us quantify our uncertainty.

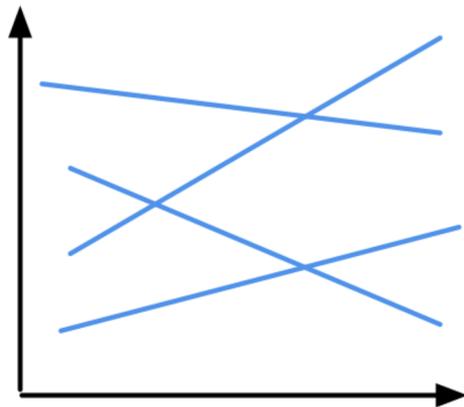
$$y | \mathbf{x} \sim \mathcal{N}(\mathbf{w}^T \psi(\mathbf{x}), \sigma^2)$$

# Bayesian Linear Regression

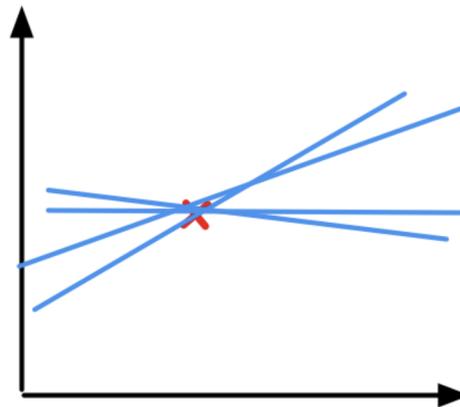
- **Prior distribution:**  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{S})$
- **Likelihood:**  $y \mid \mathbf{x}, \mathbf{w} \sim \mathcal{N}(\mathbf{w}^\top \psi(\mathbf{x}), \sigma^2)$
- Assuming fixed/known  $\mathbf{S}$  and  $\sigma^2$  is a big assumption. More on this later.

# Bayesian Linear Regression

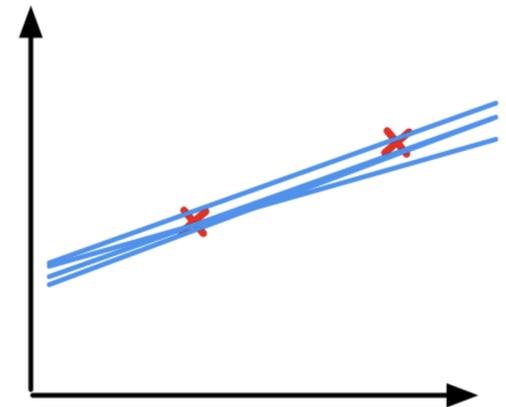
- Bayesian linear regression considers various plausible explanations for how the data were generated.
- It makes predictions using all possible regression weights, weighted by their posterior probability.
- Here are samples from the prior  $p(\mathbf{w})$  and posteriors  $p(\mathbf{w} | \mathcal{D})$



no observations



one observation



two observations

# Bayesian Linear Regression: Posterior

- Deriving the posterior distribution:

$$\begin{aligned}\log p(\mathbf{w} \mid \mathcal{D}) &= \log p(\mathbf{w}) + \log p(\mathcal{D} \mid \mathbf{w}) + \text{const} \\ &= -\frac{1}{2} \mathbf{w}^\top \mathbf{S}^{-1} \mathbf{w} - \frac{1}{2\sigma^2} \|\boldsymbol{\Psi} \mathbf{w} - \mathbf{y}\|^2 + \text{const} \\ &= -\frac{1}{2} \mathbf{w}^\top \mathbf{S}^{-1} \mathbf{w} - \frac{1}{2\sigma^2} \left( \mathbf{w}^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi} \mathbf{w} - 2\mathbf{y}^\top \boldsymbol{\Psi} \mathbf{w} + \mathbf{y}^\top \mathbf{y} \right) + \text{const} \\ &= -\frac{1}{2} \mathbf{w}^\top \left( \sigma^{-2} \boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \mathbf{S}^{-1} \right) \mathbf{w} + \frac{1}{\sigma^2} \mathbf{y}^\top \boldsymbol{\Psi} \mathbf{w} + \text{const} \\ &= -\frac{1}{2} \mathbf{w}^\top \frac{1}{\sigma^2} \left( \boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \sigma^2 \mathbf{S}^{-1} \right) \mathbf{w} + \frac{1}{\sigma^2} \mathbf{y}^\top \boldsymbol{\Psi} \mathbf{w} + \text{const} \text{ (complete the square!)}\end{aligned}$$

Thus  $\mathbf{w} \mid \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  where

$$\boldsymbol{\mu} = \left( \boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \sigma^2 \mathbf{S}^{-1} \right)^{-1} \boldsymbol{\Psi}^\top \mathbf{y}, \quad \boldsymbol{\Sigma} = \sigma^2 \left( \boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \sigma^2 \mathbf{S}^{-1} \right)^{-1}$$

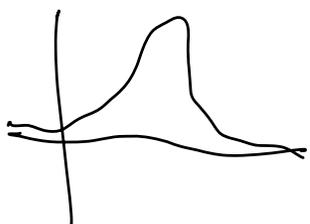
# Bayesian Linear Regression: Posterior

- Gaussian prior leads to a Gaussian posterior, and so the Gaussian distribution is the conjugate prior for linear regression model.
- Compare  $\mu = (\Psi^T \Psi + \sigma^2 \mathbf{S}^{-1})^{-1} \Psi^T \mathbf{y}$  to the closed-form solution for ridge regression:

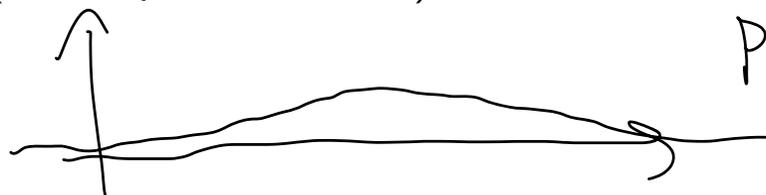
$$\mathbf{w} = (\Psi^T \Psi + \lambda \mathbf{I})^{-1} \Psi^T \mathbf{y} \rightarrow \text{Least square solution}$$

This is the mean of the posterior for  $\mathbf{S} = \left( \frac{\sigma^2}{\lambda} \mathbf{I} \right)$

- As  $\lambda \rightarrow 0$ , the standard deviation of the prior goes to  $\infty$ , and the mean of the posterior converges to the MLE (least squares solution).



$\lambda \rightarrow 0$

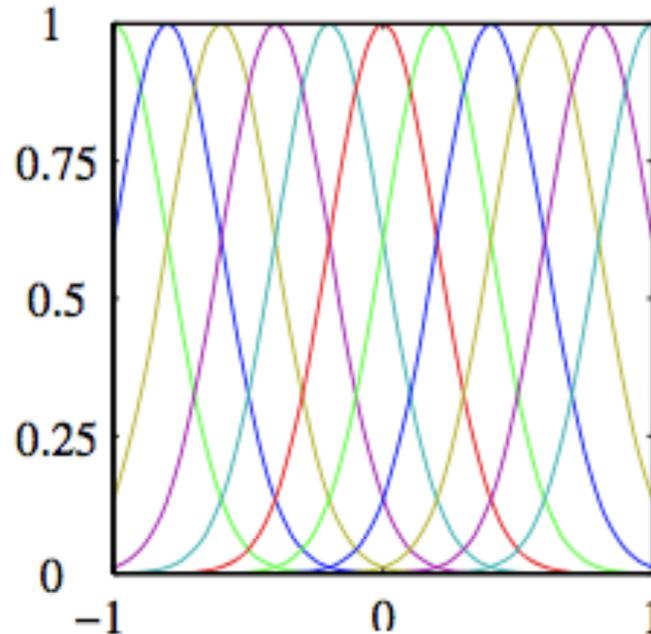


prior becomes flat  
 $\rightarrow$  all information in the data

## Radial bases example

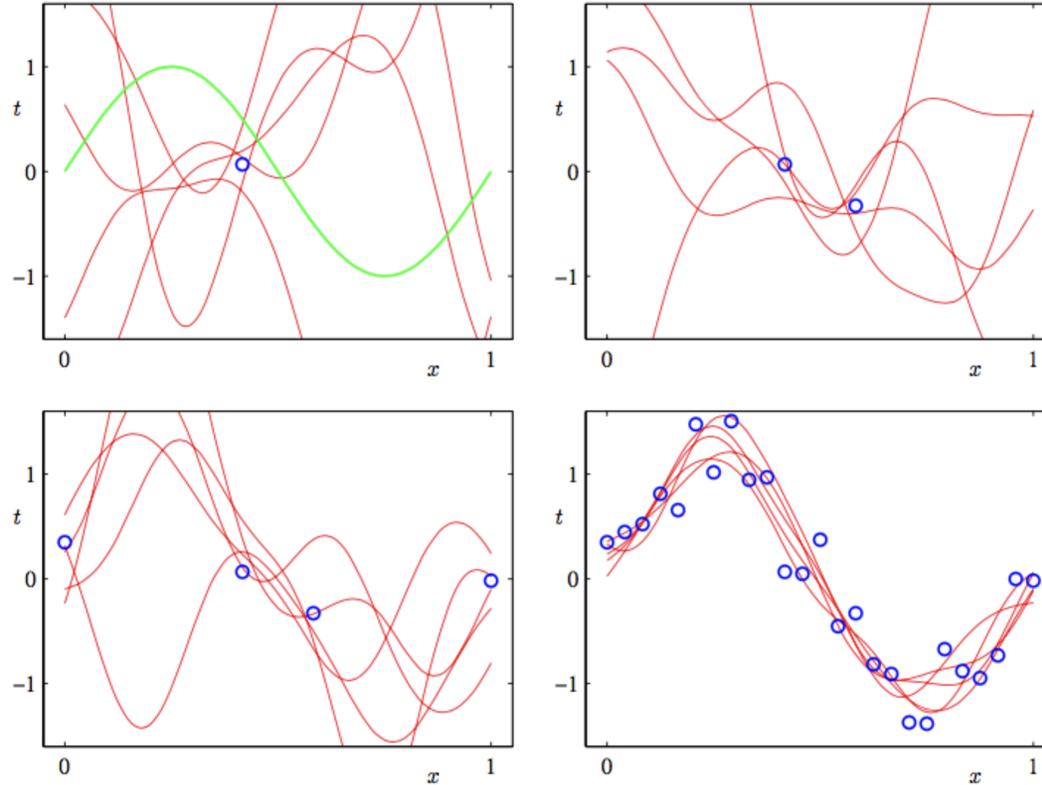
- One dimensional example:  $\{(x_i, y_i)\}_{i=1}^N, y = \mathbf{w}^\top \psi(x) + \epsilon$ .
- We use radial basis function (RBF) features

$$\psi_j(x) = \exp\left(-\frac{(x - \mu_j)^2}{2s^2}\right)$$



# Radial bases example

Functions sampled from the posterior:



## Posterior predictive distribution

- The posterior gives us distribution over the parameter space, but if we want to make predictions, the natural choice is to use the posterior predictive distribution.
- Posterior predictive distribution:

$$p(y | \mathbf{x}, \mathcal{D}) = \int \underbrace{p(y | \mathbf{x}, \mathbf{w})}_{\mathcal{N}(y; \mathbf{w}^\top \psi(\mathbf{x}), \sigma^2)} \underbrace{p(\mathbf{w} | \mathcal{D})}_{\mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})} d\mathbf{w}$$

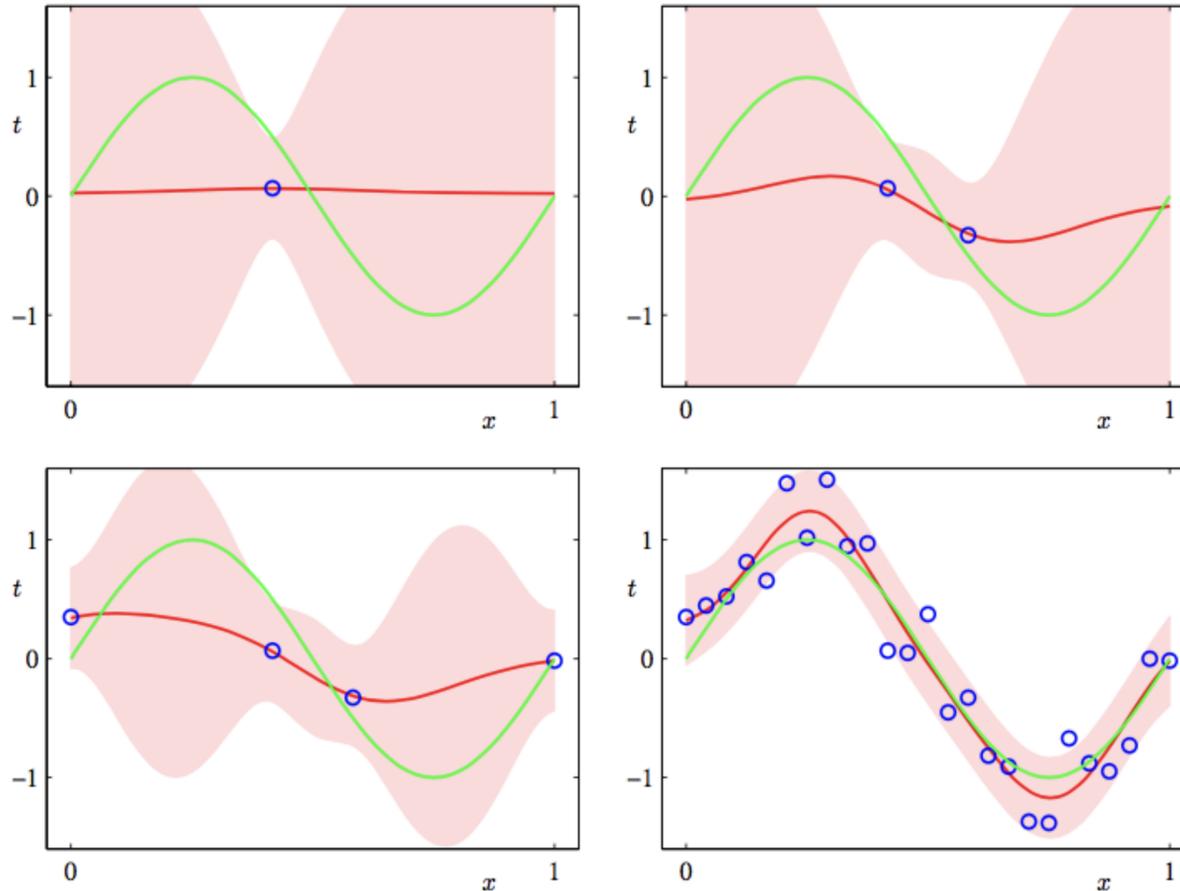
- Another interpretation:  $y = \mathbf{w}^\top \psi(\mathbf{x}) + \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  is independent of  $\mathbf{w} | \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ .
- Again by the fact that affine transformations of Gaussian vectors are Gaussian,  $y$  is a Gaussian distribution with parameters

$$\begin{aligned} \mu_{\text{pred}} &= \boldsymbol{\mu}^\top \psi(\mathbf{x}) \\ \sigma_{\text{pred}}^2 &= \underbrace{\psi(\mathbf{x})^\top \boldsymbol{\Sigma} \psi(\mathbf{x})}_{\in \mathbb{R}} + \sigma^2 \end{aligned}$$

- Hence, the posterior predictive distribution is  $\mathcal{N}(y | \mu_{\text{pred}}, \sigma_{\text{pred}}^2)$ .

# Bayesian Linear Regression

We visualize confidence intervals based on the posterior predictive distribution at each point:



## Some problems with this formulation

- The MLE will not be uniquely defined if  $N < M$ .
  - We can use ridge regression or other regularization.
- Flexibility may require a large number  $M$  of features, which may need to depend on  $N$ .
- We would like to have a method that is more automatic.
- Kernel methods and Gaussian Processes in particular offer such a flexible framework.

# Kernels

---

## Kernels: Formal definition

- A symmetric matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is **positive semidefinite** (PSD) if for every vector  $\mathbf{u} \in \mathbb{R}^N$

$$\mathbf{u}^\top \mathbf{A} \mathbf{u} \geq 0.$$

### Definition: Kernel function (Schoenberg 1938)

A **kernel**  $k(\mathbf{x}, \mathbf{x}')$  is any function such that for any  $N \geq 1$  and for any data points  $\mathbf{x}^{(i)}$  for  $i = 1, \dots, N$ , the kernel matrix  $\mathbf{K} \in \mathbb{R}^{N \times N}$  with entries  $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$  is PSD.

kernel Gram matrix  $\sim \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle$

- We can use feature maps  $\psi : \mathbb{R}^D \rightarrow \mathbb{R}^M$  to define kernels:

$$k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^\top \psi(\mathbf{x}').$$

- Feature maps define kernels but not all kernels are like that (this can be generalized to “infinite dimensional” feature maps).

$\Rightarrow$  kernels generalize inner products

## Feature map defines a kernel

- Let  $k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^\top \psi(\mathbf{x}')$
- The kernel matrix is given as  $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ ,  $\mathbf{K} = \Psi\Psi^\top$ .
- We show that this matrix is positive semi-definite,  $\forall \mathbf{u} \in \mathbb{R}^N$ ,

$$\mathbf{u}^\top \mathbf{K} \mathbf{u} = \mathbf{u}^\top \Psi \Psi^\top \mathbf{u} = (\Psi^\top \mathbf{u})^\top \Psi^\top \mathbf{u} = \|\Psi^\top \mathbf{u}\|^2 \geq 0.$$

Main points:

- Forget the feature map.  $\rightarrow$  instead of taking the feature map then inner product, use the kernel
- We can directly choose a kernel and work with it!
- The dimension of the feature space does not matter anymore.
- Kernels provide a measure of proximity between  $\mathbf{x}$  and  $\mathbf{x}'$ .

## Kernels: Examples

Example 1:

- $D$ -dimensional inputs:  $\mathbf{x} = (x_1, x_2, \dots, x_D)^\top$  and  $\mathbf{z} = (z_1, z_2, \dots, z_D)^\top$

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^\top \mathbf{z})^2 = (x_1 z_1 + x_2 z_2 + \dots)^2 = \sum_{i,j=1}^D x_i z_i x_j z_j \\ &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 + \dots \\ &= (x_1^2, x_2^2, \dots, \sqrt{2}x_1 x_2, \dots)^\top (z_1^2, z_2^2, \dots, \sqrt{2}z_1 z_2, \dots) \\ &= \psi(\mathbf{x})^\top \psi(\mathbf{z}) \end{aligned}$$

$D^2$  terms  
↗

$\in \mathbb{R}^D$

lengthscale

Example 2 (Gaussian kernel):  $k(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2 / 2\ell^2)$ .

- The feature vector has infinite dimension here! (a bit of functional analysis)

# Constructing kernels from kernels

Given valid kernels  $k_1(\mathbf{x}, \mathbf{x}')$  and  $k_2(\mathbf{x}, \mathbf{x}')$ , the following kernels will also be valid:

$$k(\mathbf{x}, \mathbf{x}') = ck_1(\mathbf{x}, \mathbf{x}') \quad \text{for } c > 0,$$

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})k_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}')$$

$$\langle A^{1/2} \mathbf{x}, A^{1/2} \mathbf{x}' \rangle \iff k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A} \mathbf{x}' \quad (\mathbf{A} \text{ PSD})$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}'))$$

$$k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}'))$$

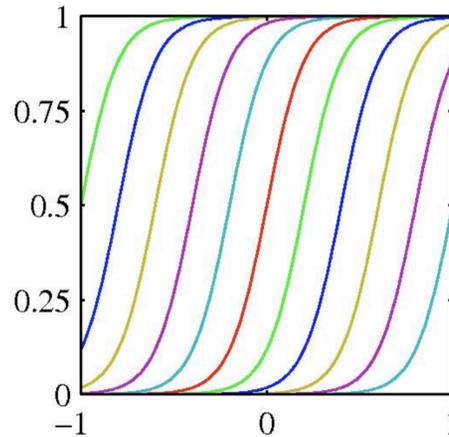
where  $q$  polynomial with  $\geq 0$  coefficients.

# Radial basis functions

To get a better feeling for these methods consider the case where kernel is defined by a radial basis function.

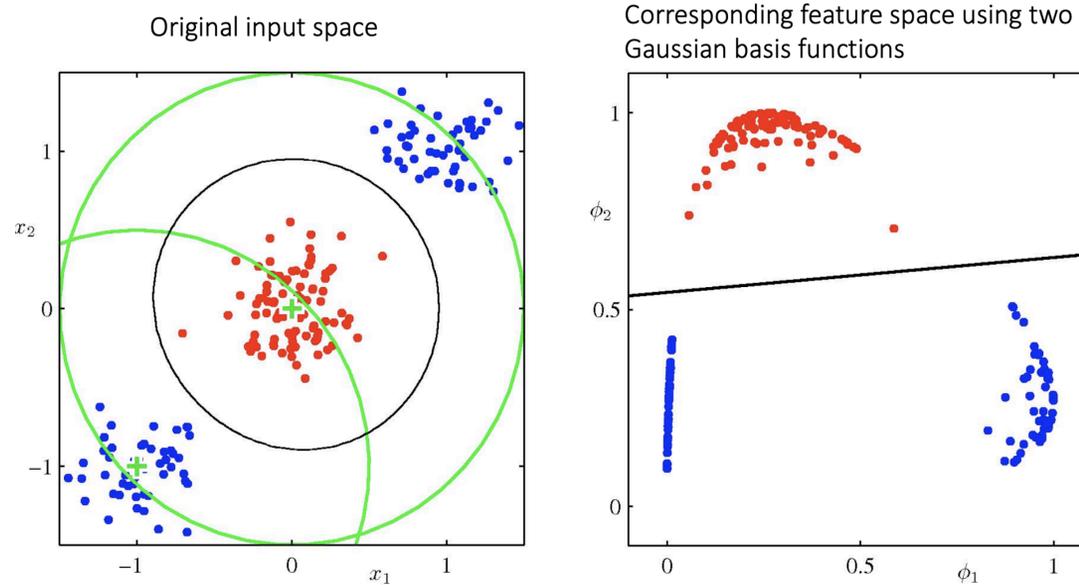
- Radial basis functions depend only on the distance from  $\mu_j$ , i.e.

$$\psi_j(\mathbf{x}) = h(\|\mathbf{x} - \mu_j\|).$$



- Sigmoidal basis functions:  $h$  is sigmoid.
- Gaussian basis functions:  $h$  is normal pdf  $\rightarrow e^{-\frac{\|\mathbf{x} - \mu_j\|^2}{2\sigma^2}}$

# Example: Radial basis functions



- We define two Gaussian basis functions with centers shown by the green crosses, and with contours shown by the green circles.
- Linear decision boundary (right) corresponds to the nonlinear decision boundary in the input space (left, black curve).

# Gaussian Processes

---

# Bayesian Linear Regression

- We gave linear regression a probabilistic interpretation by assuming a Gaussian noise model:

$$y \mid \mathbf{x} \sim \mathcal{N}(\hat{y}(\mathbf{x}), \sigma^2), \quad \hat{y}(\mathbf{x}) = \mathbf{w}^\top \psi(\mathbf{x})$$

- and a Gaussian prior

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \frac{1}{\alpha} \mathbf{I}_M)$$

**The prior induces a probability distribution over  $\hat{y}$**

$$\hat{y} = \Psi \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \frac{1}{\alpha} \Psi \Psi^\top)$$

Indeed:  $\mathbb{E}(\Psi \mathbf{w}) = \Psi \mathbb{E}(\mathbf{w}) = \mathbf{0}$  and  $\text{var}(\Psi \mathbf{w}) = \mathbb{E}(\Psi \mathbf{w} \mathbf{w}^\top \Psi^\top) = \Psi \mathbb{E}(\mathbf{w} \mathbf{w}^\top) \Psi^\top = \frac{1}{\alpha} \Psi \Psi^\top$ .

# Distribution over prediction function

- In practice, we evaluate the prediction function  $\hat{y}(\mathbf{x})$  at specific points, for example at the training data points  $\mathbf{x}^{(i)}$  for  $i = 1, \dots, N$ .
- So we are interested in the joint distribution of the function values

$$\hat{y}(\mathbf{x}^{(1)}), \dots, \hat{y}(\mathbf{x}^{(N)})$$

which we denote by the vector  $\hat{\mathbf{y}} = (\hat{y}(\mathbf{x}^{(1)}), \dots, \hat{y}(\mathbf{x}^{(N)}))$ .

- We showed that

$$\hat{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}) \quad \mathbf{K} = \frac{1}{\alpha} \boldsymbol{\Psi} \boldsymbol{\Psi}^\top$$

given by  $\langle \psi(\mathbf{x}^{(i)}), \psi(\mathbf{x}^{(j)}) \rangle$

where  $\mathbf{K}$  is the (scaled) Gram matrix

$$K_{ij} = \frac{1}{\alpha} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \frac{1}{\alpha} \psi(\mathbf{x}^{(i)})^\top \psi(\mathbf{x}^{(j)})$$

# Gaussian process

## Definition:

A **Gaussian process** is a probability distribution over functions  $\hat{y}(\mathbf{x})$  such that for any  $N \geq 1$  and any set of  $N$  points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$  in  $\mathbb{R}^D$ , the vector  $(\hat{y}(\mathbf{x}^{(1)}), \dots, \hat{y}(\mathbf{x}^{(N)}))$  is jointly Gaussian.

$\sim \mathcal{N}(\mu, \Sigma)$  depend on  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$

- The joint distribution is specified completely by the second-order statistics, i.e. the mean and the covariance functions.
- In most applications, the mean function of  $\hat{y}(\mathbf{x})$  can be set to zero and then the Gaussian process is completely specified by the covariance function

$$\mathbb{E}[\hat{y}(\mathbf{x})\hat{y}(\mathbf{x}')] = \frac{1}{\alpha} k(\mathbf{x}, \mathbf{x}')$$

GP  $\rightarrow \Sigma = K$  is the kernel gram matrix of  $\mathcal{R}$

# Gaussian process (GP)

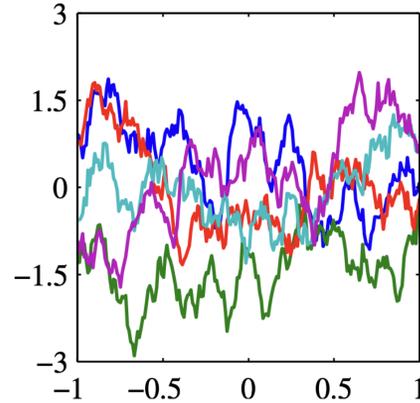
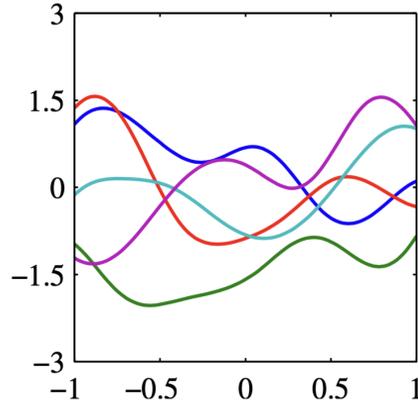
- Directly define the kernel of a Gaussian process, not worrying about the feature map.

Discretize the input space  $x^{(1)}, \dots, x^{(N)}$

• compute  $K$

• output

$$y \sim \mathcal{N}(0, K)$$
$$f_* \sim \mathcal{N}(0, \mathbb{I}_N)$$



Samples from GP for a Gaussian kernel  $v^2 e^{-\|x^{(i)} - x^{(j)}\|_2^2 / (2\ell^2)}$  (left) and an exponential kernel  $v^2 e^{-\|x^{(i)} - x^{(j)}\|_2 / (2\ell)}$  (right).

(How do you think these plots are generated?)

# Gaussian processes for regression: what we learn from the data

- We have the linear model

$$y \mid \mathbf{x} \sim \mathcal{N}(\hat{y}(\mathbf{x}), \sigma^2) \quad \hat{y}(\mathbf{x}) = \mathbf{w}^\top \psi(\mathbf{x})$$

- Given  $N$  independent observations, we have

$$\mathbf{y} \mid \hat{\mathbf{y}} \sim \mathcal{N}(\hat{\mathbf{y}}, \sigma^2 \mathbf{I}_N), \quad \hat{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \quad \mathbf{K} = \frac{1}{\alpha} \boldsymbol{\Psi} \boldsymbol{\Psi}^\top.$$

- Therefore the marginal of  $\mathbf{y}$  is given by

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \quad \mathbf{C} = \mathbf{K} + \sigma^2 \mathbf{I}_N$$

where the corresponding kernel is

$$c(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \frac{1}{\alpha} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \sigma^2 \delta(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

$\delta(\mathbf{x}, \mathbf{x}') = 1$  if  $\mathbf{x} = \mathbf{x}'$  and  $\delta(\mathbf{x}, \mathbf{x}') = 0$  otherwise.

# Gaussian processes for regression: predictive distributions

- Denote now  $\mathbf{y}_N = (y^{(1)}, y^{(2)}, \dots, y^{(N)})$ .

- We have the marginal of  $\mathbf{y}_N$  given by

$$\mathbf{y}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_N) \quad \mathbf{C}_N = \mathbf{K}_N + \sigma^2 \mathbf{I}_N.$$

- This reflects the two Gaussian sources of randomness.

**Goal:** We want to predict for a new output  $y^{(N+1)}$  given a new input  $\mathbf{x}^{(N+1)}$ .

- We need

$$p(y^{(N+1)} \mid \mathbf{y}_N)$$

- Note that  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}, \mathbf{x}^{(N+1)}$  are treated as constants.

# Gaussian processes for regression: predictive distributions

$$\left( \underbrace{y^{(1)}, \dots, y^{(N)}}_{\text{training set}}, y^{(N+1)} \right)$$

- We have

$$\mathbf{y}_{N+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{N+1}) \quad \mathbf{C}_{N+1} = \mathbf{K}_{N+1} + \sigma^2 \mathbf{I}_{N+1}$$

where

$$\mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^\top & c \end{bmatrix} = \begin{pmatrix} k(x^{(1)}, x^{(N+1)}) \\ \vdots \\ k(x^{(N)}, x^{(N+1)}) \\ c \end{pmatrix}$$

- Here,  $c = \frac{1}{\alpha} k(\mathbf{x}^{(N+1)}, \mathbf{x}^{(N+1)}) + \sigma^2$
- $\mathbf{k}$  is a vector with entries  $k_i = \frac{1}{\alpha} k(\mathbf{x}^{(i)}, \mathbf{x}^{(N+1)})$

- Since the vector  $\mathbf{y}_{N+1}$  is Gaussian, we easily find  $y^{(N+1)} \mid \mathbf{y}_N$ .

# Property of Multivariate Gaussian Distribution

Recall:

- If we have  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

- Then,

$$\mathbf{x}_2 \mid (\mathbf{x}_1 = \mathbf{a}) \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$$

with

$$\mathbf{m} = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{a} - \boldsymbol{\mu}_1), \quad \mathbf{C} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}.$$

# Gaussian processes for regression

Recall:

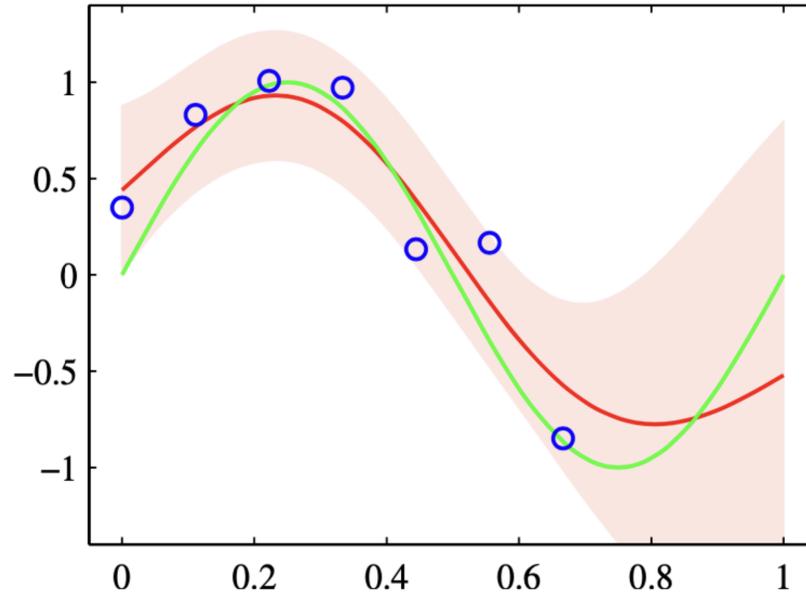
$$\mathbf{y}_{N+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{N+1}), \quad \mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^\top & c \end{bmatrix}.$$

- Since  $\mathbf{y}_{N+1}$  is multivariate Gaussian,  $y^{(N+1)} \mid \mathbf{y}_N$  is also Gaussian with mean and variance

$$\text{mean} = \mathbf{k}^\top \mathbf{C}_N^{-1} \mathbf{y}_N \quad \text{variance} = c - \mathbf{k}^\top \mathbf{C}_N^{-1} \mathbf{k}$$

- These are the key results that define Gaussian process regression.
- The vector  $\mathbf{k}$  is a function of the new test input  $\mathbf{x}^{(N+1)}$ .
- The predictive distribution is a Gaussian whose mean and variance both depend on  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}, \mathbf{x}^{(N+1)}$ .

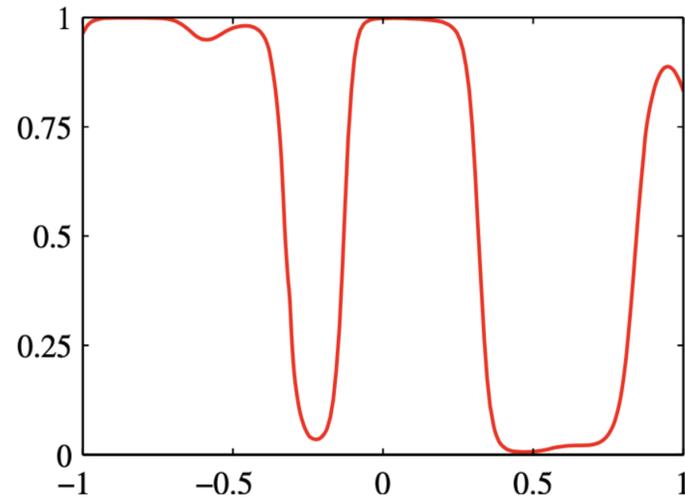
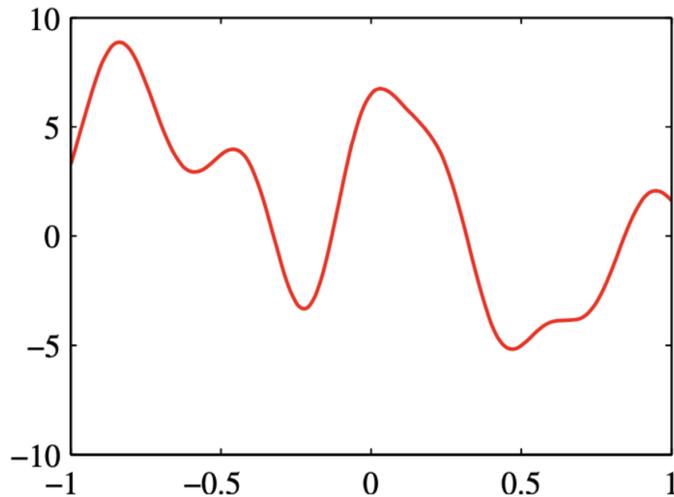
## GPs for regression



- The green curve is the true sinusoidal function from which the data points, shown in blue, are obtained.
- The red line shows the **mean** of the Gaussian process predictive distribution.
- The shaded region corresponds to plus and minus two standard deviations.

## GPs for classification

- Consider a classification problem with target variables  $y \in \{0, 1\}$
- We define a Gaussian process over a function  $a(\mathbf{x})$  and then transform the function using sigmoid  $\hat{y}(\mathbf{x}) = \sigma(a(\mathbf{x}))$ .
- We obtain a non-Gaussian stochastic process over functions  $\hat{y}(\mathbf{x}) \in (0, 1)$ .



Left:  $a(\mathbf{x})$  Right:  $\hat{y}(\mathbf{x})$

## GPs for classification

- The probability distribution over target is then given by

$$p(y|a) = \sigma(a)^y (1 - \sigma(a))^{1-y}, \quad y \in \{0, 1\}.$$

- We need to compute

$$p(y^{(N+1)} | \mathbf{y}_N)$$

and notice that  $a(\mathbf{x})$  is a Gaussian process but  $\hat{y}(\mathbf{x})$  is not.

- We have  $\mathbf{a}_{N+1} \sim \mathcal{N}(0, \mathbf{C}_{N+1})$ , where

$$C_{N+1}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \frac{1}{\alpha} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}).$$

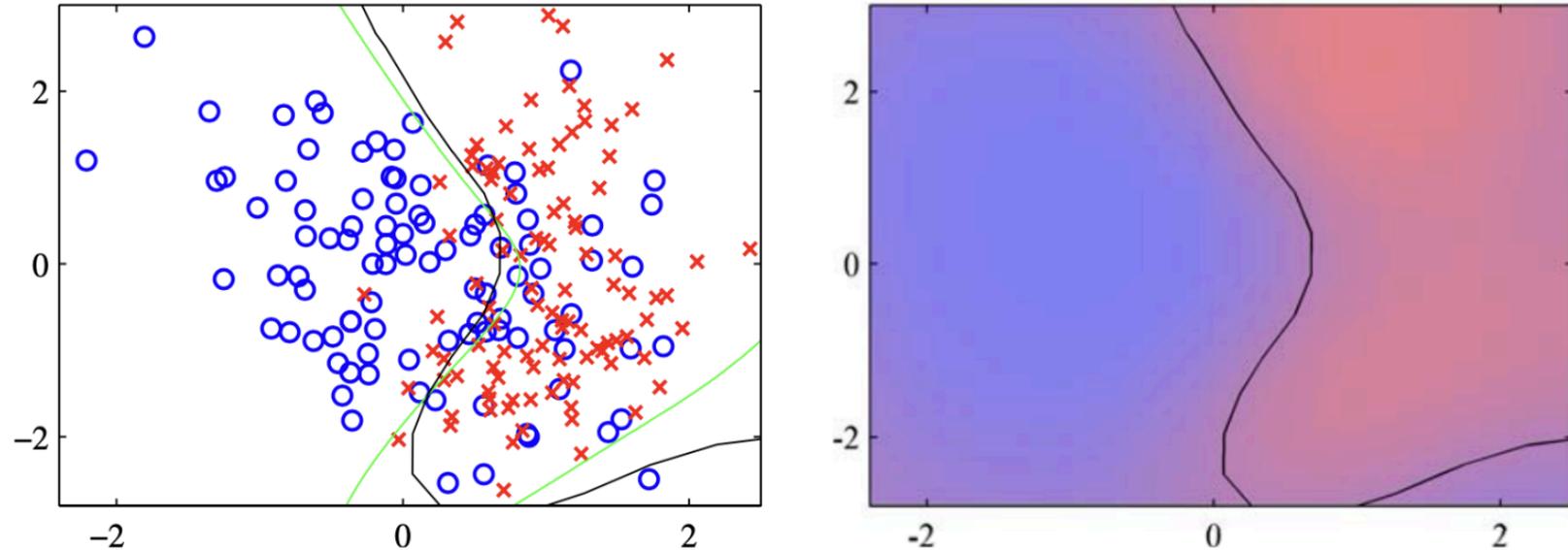
- But  $\mathbf{a}_N$  is not observed, so we write

$$p(y^{(N+1)} | \mathbf{y}_N) = \int p(y^{(N+1)} | \mathbf{a}_{N+1}) p(\mathbf{a}_{N+1} | \mathbf{y}_N) d\mathbf{a}_{N+1}$$

- This is intractable. We need MCMC based methods, or numerical integration to approximate this integral.

# GPs for classification: Illustration

- Illustration of GPs for classification:



- Left: optimal decision boundary from the true distribution in green, and the decision boundary from the Gaussian process classifier in black.
- Right: predicted posterior for the blue and red classes together with the Gaussian process decision boundary.

# Learning the hyperparameters

- We didn't do any learning other than choosing a kernel!
- Rather than fixing the covariance function  $\frac{1}{\alpha}k(\mathbf{x}, \mathbf{x}')$ , we may prefer to use a parametric family of functions and then infer the parameter values from the data.
- Denoting the hyperparameters with  $\theta$ , one can easily write down the likelihood of the Gaussian process model.

$$\log p(\mathbf{y} | \theta) = -\frac{1}{2} \log |\mathbf{C}_N| - \frac{1}{2} \mathbf{y}^\top \mathbf{C}_N^{-1} \mathbf{y} - \frac{N}{2} \log(2\pi)$$

- The next step is standard: gradient based optimization, grid search etc.

→ maximize likelihood w.r.t.  $\theta$