

# From Gaussian variables to Gaussian processes

Mauricio A. Álvarez

Reasoning and Learning under Uncertainty  
The University of Manchester



The University of Manchester

# Gaussian processes in machine learning

- Gaussian processes (GPs) or Gaussian random fields.
- They were introduced by George Matheron in 1960 under the name of **kriging** (geostatistics literature).
- Well known in the Statistics and Probability communities.
- Growing in popularity in machine learning since the 90s.

# Gaussian processes in machine learning

- A Gaussian process generalises the multivariate Gaussian distribution to the infinite dimensional setting.
- Most common application is non-linear regression.
- They have been also used in pattern classification, dimensionality reduction, multi-task learning and Bayesian optimisation.

# Contents

Univariate and multivariate Gaussian distributions

Gaussian processes

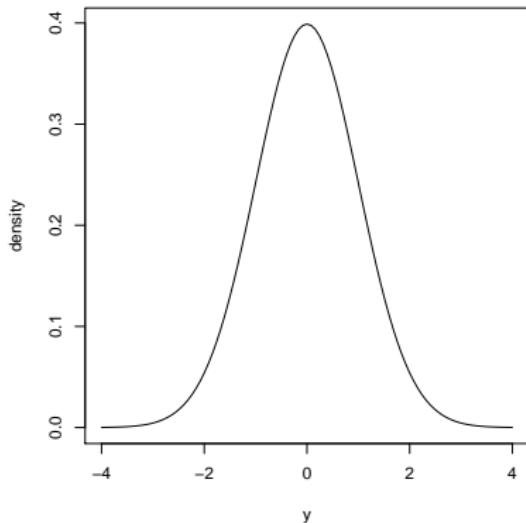
Connection to Bayesian Linear regression

Resources

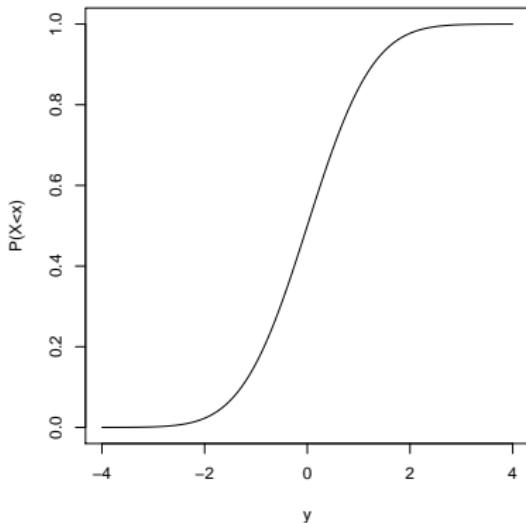
Summary

# Univariate Gaussian distributions

PDF of a  $N(0,1)$  random variable



CDF of a  $N(0,1)$  random variable



$$Y \sim N(\mu, \sigma^2)$$

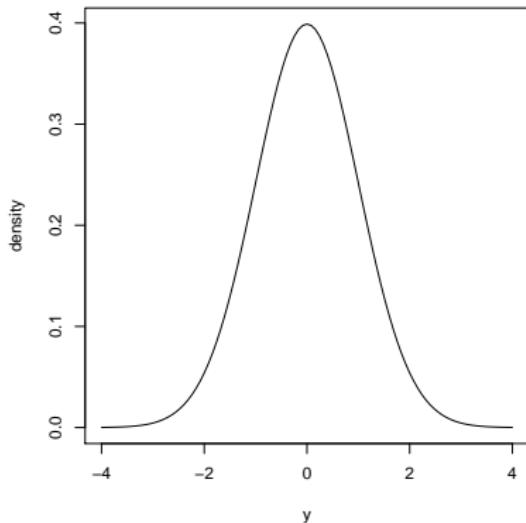
PDF:  $f_Y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$

CDF:  $F_Y(y) = \mathbb{P}(Y \leq y)$  not known in closed form

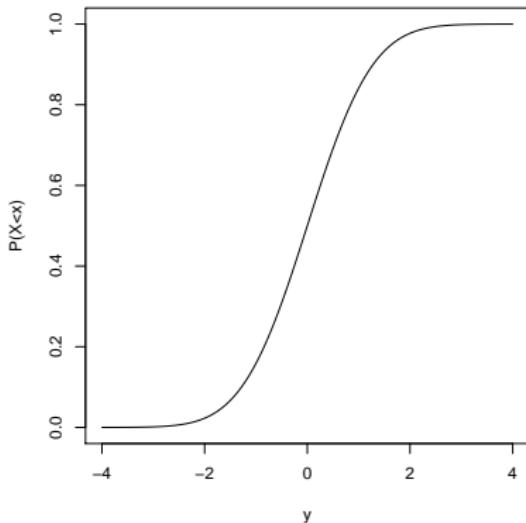
If  $Z \sim N(0, 1)$  then  $Y = \mu + \sigma Z \sim N(\mu, \sigma^2)$

# Univariate Gaussian distributions

PDF of a  $N(0,1)$  random variable



CDF of a  $N(0,1)$  random variable



$$Y \sim N(\mu, \sigma^2)$$

PDF:  $f_Y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$

CDF:  $F_Y(y) = \mathbb{P}(Y \leq y)$  not known in closed form

If  $Z \sim N(0, 1)$  then  $Y = \mu + \sigma Z \sim N(\mu, \sigma^2)$

# Univariate Gaussians

The normal/Gaussian distribution occurs naturally and is convenient mathematically

- Central limit theorem.
- Family of normal distributions is closed under linear operations.
- If  $Y$  and  $Z$  are jointly normally distributed and are uncorrelated, then they are independent.

# Univariate Gaussians

The normal/Gaussian distribution occurs naturally and is convenient mathematically

- Central limit theorem.
- Family of normal distributions is closed under linear operations.
- If  $Y$  and  $Z$  are jointly normally distributed and are uncorrelated, then they are independent.

# Univariate Gaussians

The normal/Gaussian distribution occurs naturally and is convenient mathematically

- Central limit theorem.
- Family of normal distributions is closed under linear operations.
- If  $Y$  and  $Z$  are jointly normally distributed and are uncorrelated, then they are independent.

# Univariate Gaussians

The normal/Gaussian distribution occurs naturally and is convenient mathematically

- Central limit theorem.
- Family of normal distributions is closed under linear operations.
- If  $Y$  and  $Z$  are jointly normally distributed and are uncorrelated, then they are independent.

# Multivariate Gaussian distributions

'Multivariate' = two or more random variables

Suppose  $Y \in \mathbb{R}^d$  has a multivariate Gaussian distribution with

- **mean vector**  $\mu \in \mathbb{R}^d$
- **covariance matrix**  $\Sigma \in \mathbb{R}^{d \times d}$ .

Write

$$Y \sim N_d(\mu, \Sigma)$$

Bivariate Gaussian: d=2

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{21}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

$$\text{Var}(Y_i) = \sigma_i^2 \quad \text{Cov}(Y_i, Y_j) = \rho_{ij}\sigma_i\sigma_j \quad \text{Cor}(Y_i, Y_j) = \rho_{12} \text{ for } i \neq j$$

pdf:  $f(y | \mu, \Sigma) = |\Sigma|^{-\frac{1}{2}}(2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(y - \mu)^\top \Sigma^{-1}(y - \mu)\right)$

# Multivariate Gaussian distributions

'Multivariate' = two or more random variables

Suppose  $Y \in \mathbb{R}^d$  has a multivariate Gaussian distribution with

- **mean vector**  $\mu \in \mathbb{R}^d$
- **covariance matrix**  $\Sigma \in \mathbb{R}^{d \times d}$ .

Write

$$Y \sim N_d(\mu, \Sigma)$$

Bivariate Gaussian:  $d=2$

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{21}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

$$\text{Var}(Y_i) = \sigma_i^2 \quad \text{Cov}(Y_i, Y_j) = \rho_{ij}\sigma_i\sigma_j \quad \text{Cor}(Y_i, Y_j) = \rho_{12} \text{ for } i \neq j$$

pdf:  $f(y | \mu, \Sigma) = |\Sigma|^{-\frac{1}{2}} (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(y - \mu)^\top \Sigma^{-1}(y - \mu)\right)$

# Multivariate Gaussian distributions

'Multivariate' = two or more random variables

Suppose  $Y \in \mathbb{R}^d$  has a multivariate Gaussian distribution with

- **mean vector**  $\mu \in \mathbb{R}^d$
- **covariance matrix**  $\Sigma \in \mathbb{R}^{d \times d}$ .

Write

$$Y \sim N_d(\mu, \Sigma)$$

**Bivariate Gaussian: d=2**

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{21}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

$$\text{Var}(Y_i) = \sigma_i^2 \quad \text{Cov}(Y_i, Y_j) = \rho_{ij}\sigma_i\sigma_j \quad \text{Cor}(Y_i, Y_j) = \rho_{12} \text{ for } i \neq j$$

pdf:  $f(y | \mu, \Sigma) = |\Sigma|^{-\frac{1}{2}} (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(y - \mu)^\top \Sigma^{-1}(y - \mu)\right)$

# Multivariate Gaussian distributions

'Multivariate' = two or more random variables

Suppose  $Y \in \mathbb{R}^d$  has a multivariate Gaussian distribution with

- **mean vector**  $\mu \in \mathbb{R}^d$
- **covariance matrix**  $\Sigma \in \mathbb{R}^{d \times d}$ .

Write

$$Y \sim N_d(\mu, \Sigma)$$

**Bivariate Gaussian: d=2**

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{21}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

$$\text{Var}(Y_i) = \sigma_i^2 \quad \text{Cov}(Y_i, Y_j) = \rho_{ij}\sigma_i\sigma_j \quad \text{Cor}(Y_i, Y_j) = \rho_{12} \text{ for } i \neq j$$

pdf:  $f(y | \mu, \Sigma) = |\Sigma|^{-\frac{1}{2}} (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(y - \mu)^\top \Sigma^{-1}(y - \mu)\right)$

# Multivariate Gaussian distributions

'Multivariate' = two or more random variables

Suppose  $Y \in \mathbb{R}^d$  has a multivariate Gaussian distribution with

- **mean vector**  $\mu \in \mathbb{R}^d$
- **covariance matrix**  $\Sigma \in \mathbb{R}^{d \times d}$ .

Write

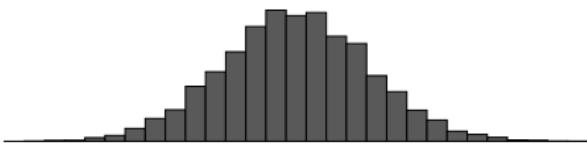
$$Y \sim N_d(\mu, \Sigma)$$

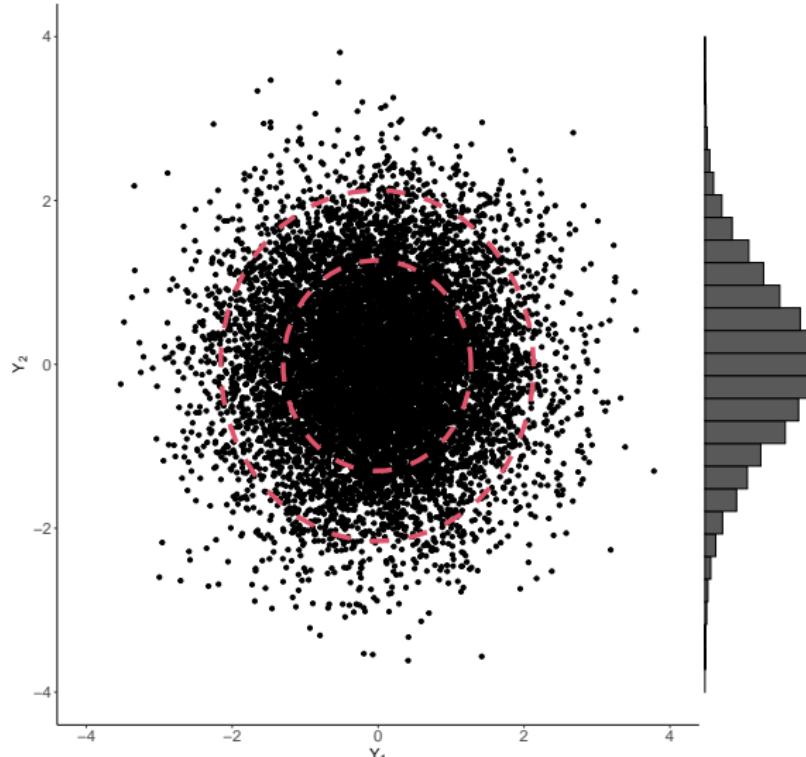
**Bivariate Gaussian: d=2**

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{21}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

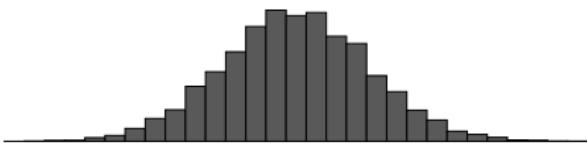
$$\text{Var}(Y_i) = \sigma_i^2 \quad \text{Cov}(Y_i, Y_j) = \rho_{ij}\sigma_i\sigma_j \quad \text{Cor}(Y_i, Y_j) = \rho_{12} \text{ for } i \neq j$$

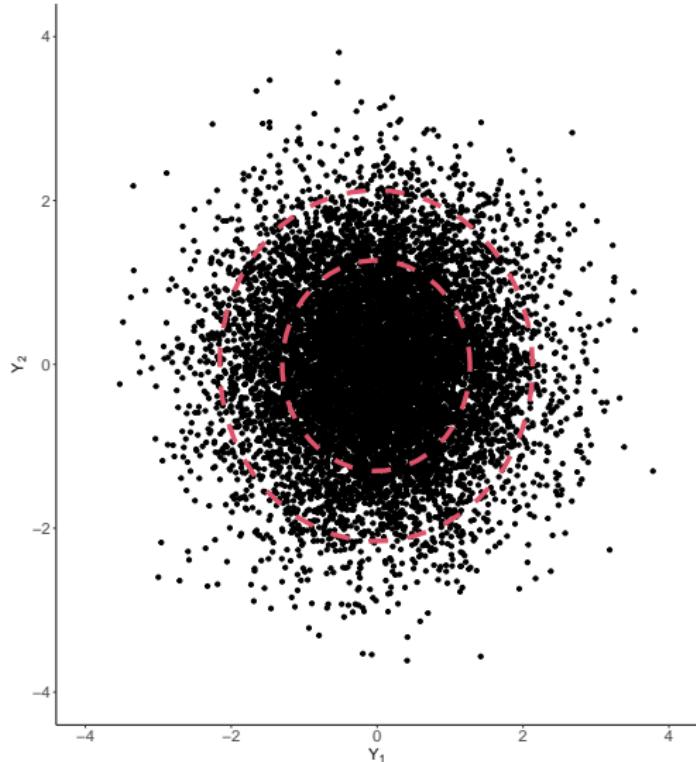
pdf:  $f(y | \mu, \Sigma) = |\Sigma|^{-\frac{1}{2}}(2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(y - \mu)^\top \Sigma^{-1}(y - \mu)\right)$


$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$


$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

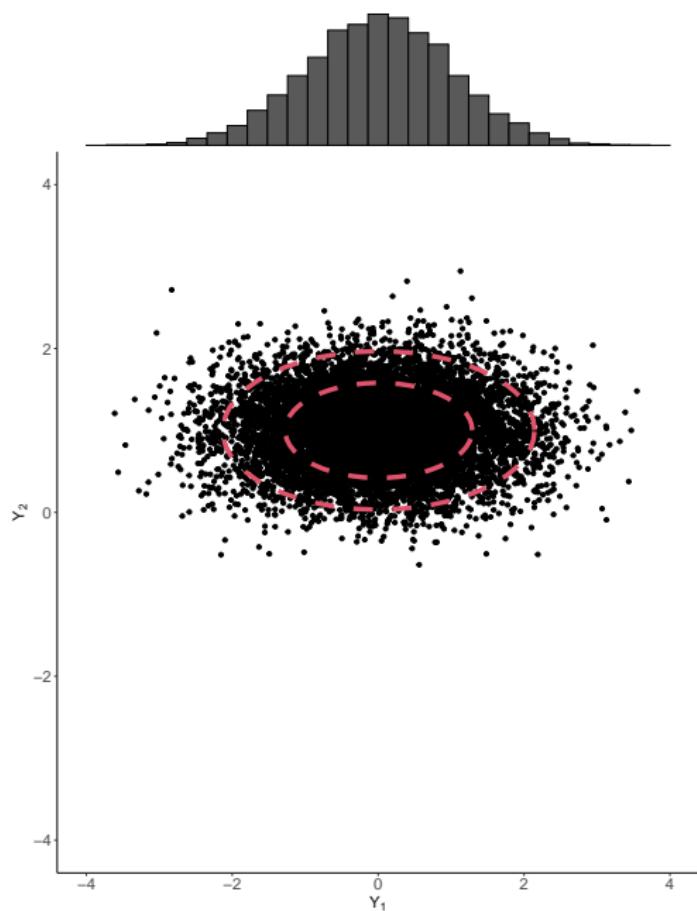
So  $\text{Cor}(Y_1, Y_2) = 0$   
hence  $Y_1$   
independent of  $Y_2$


$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$



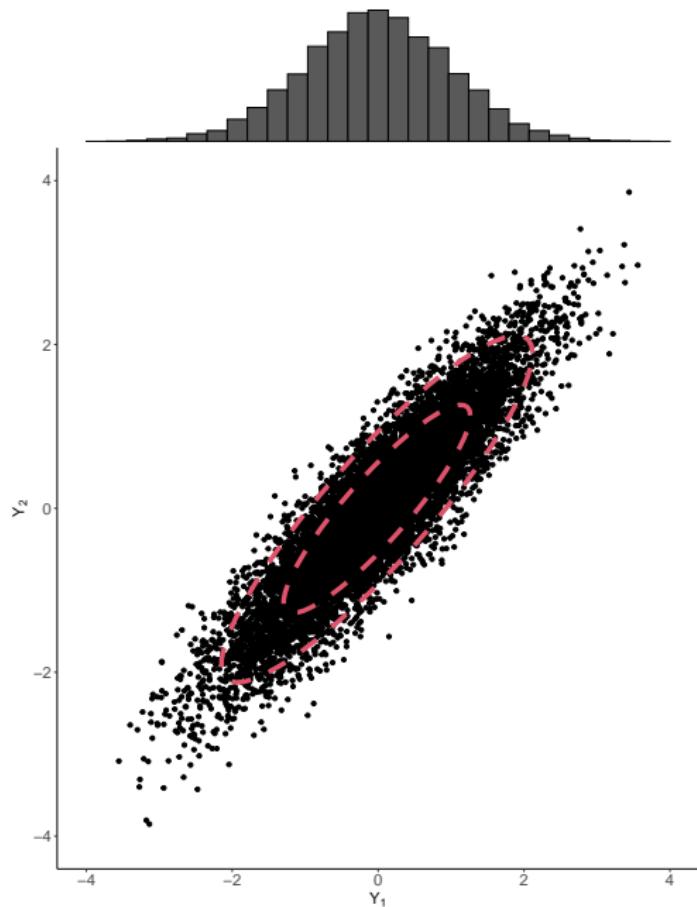

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

So  $\text{Cor}(Y_1, Y_2) = 0$   
hence  $Y_1$   
independent of  $Y_2$



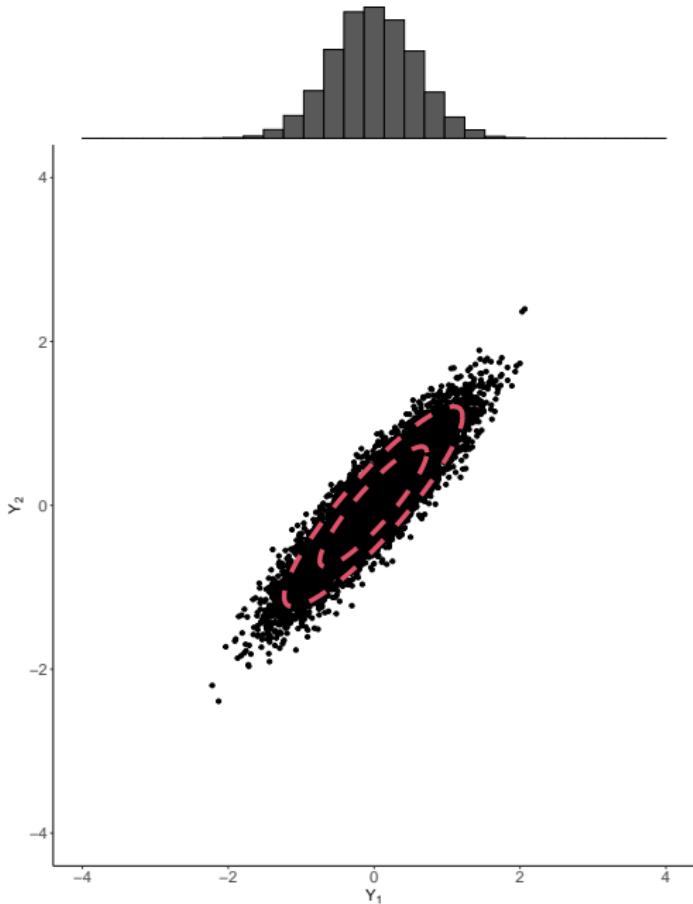
$$\mu = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 0.2 \end{pmatrix}$$



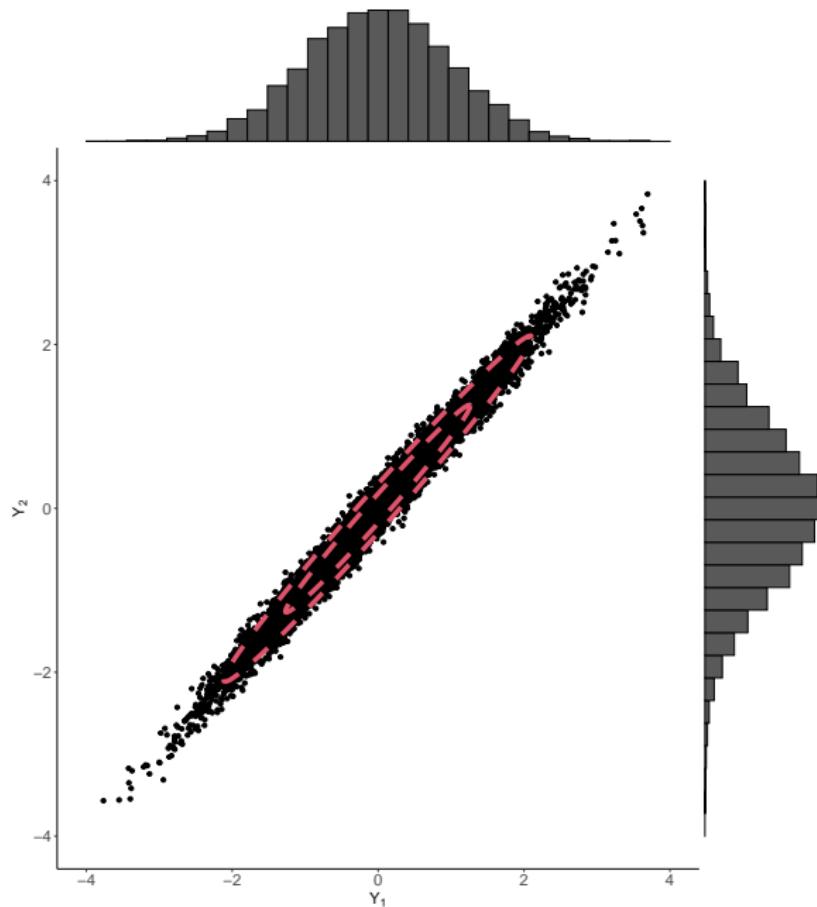
$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$



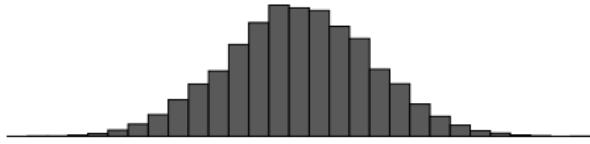
$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \frac{1}{3} \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$



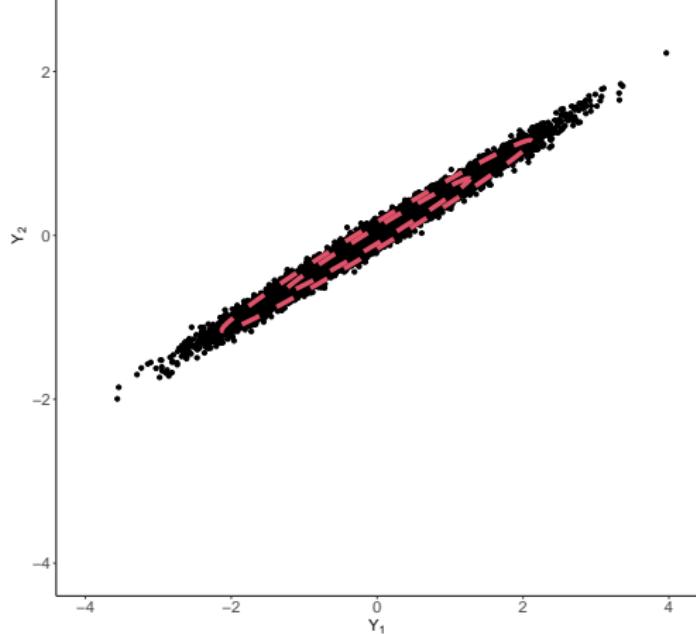
$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}$$

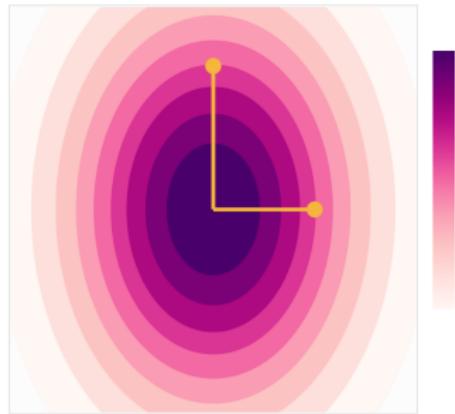

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.54 \\ 0.54 & 0.3 \end{pmatrix}$$

$$Cor(Y_1, Y_2) = \\ 0.54 / \sqrt{0.3} = \\ 0.99$$



# Visual exploration



Covariance matrix ( $\Sigma$ )

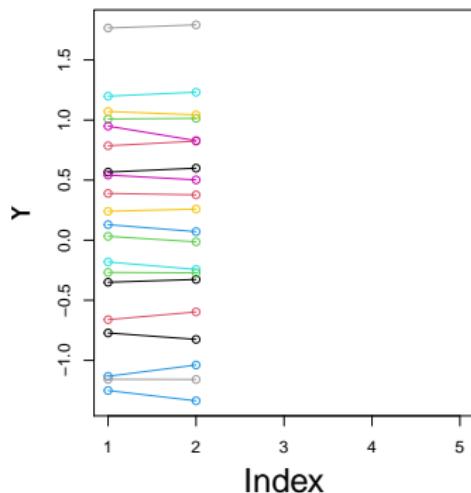
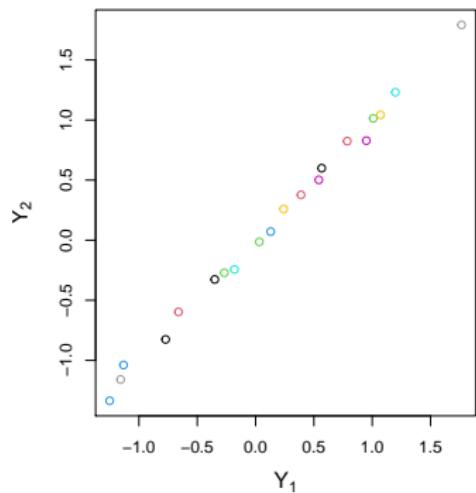
1	0.7
0.7	2

By dragging the handles you can adjust the variance along each dimension, as well as the correlation between the two random variables. *Violet* values show a high probability inside the distribution.

Taken from: "Visual exploration of Gaussian processes" by J Götler, R Kehlbeck and O Deussen (2019)

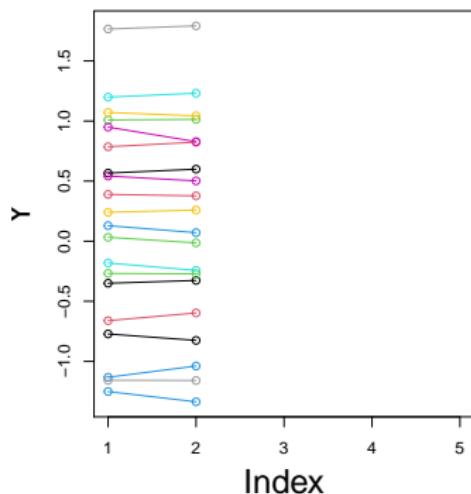
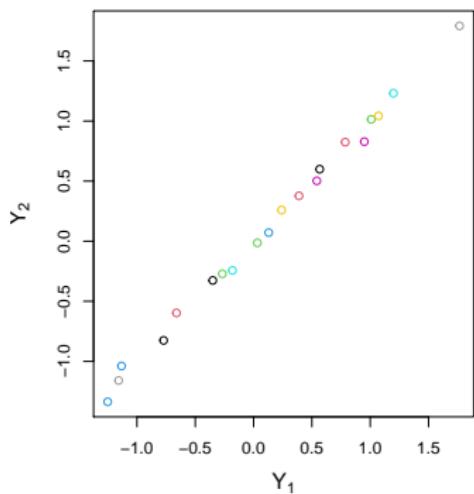
# Visualisation in more than two dimensions

Hard to visualise in dimensions  $> 2$ , so stack points next to each other.  
So for 2d instead of we have



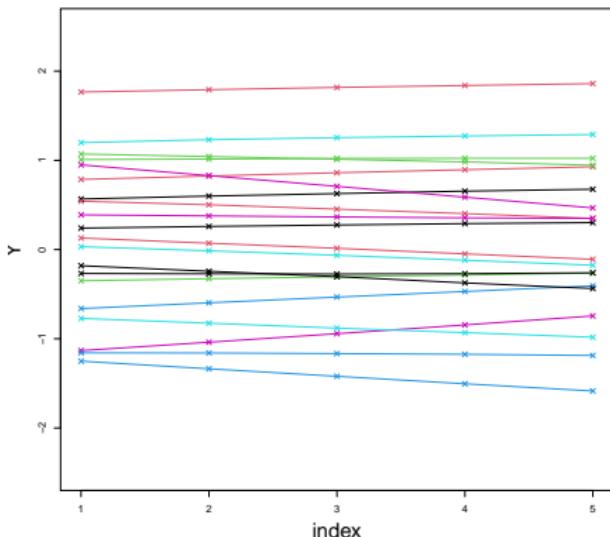
# Visualisation in more than two dimensions

Hard to visualise in dimensions  $> 2$ , so stack points next to each other.  
So for 2d instead of we have



# Consider $d = 5$

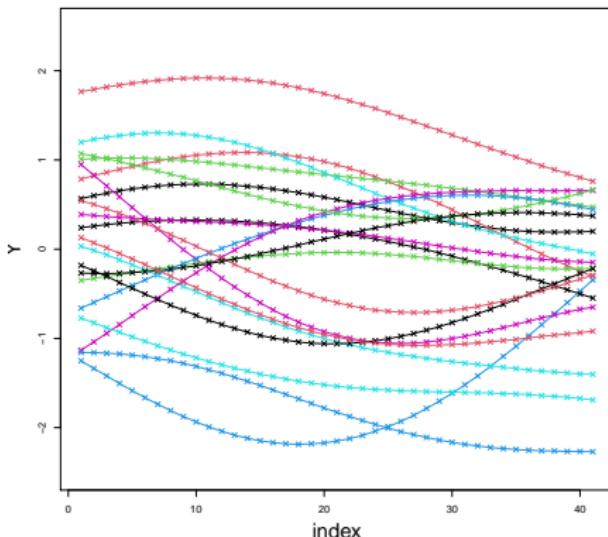
$$\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.99 & 0.98 & 0.97 & 0.96 \\ 0.99 & 1 & 0.99 & 0.98 & 0.97 \\ 0.98 & 0.99 & 1 & 0.99 & 0.98 \\ 0.97 & 0.98 & 0.99 & 1 & 0.99 \\ 0.96 & 0.97 & 0.98 & 0.99 & 1 \end{pmatrix}$$



Each line is one sample.

# Consider $d = 50$

$$\mu = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.99 & 0.98 & 0.97 & 0.96 & \dots \\ 0.99 & 1 & 0.99 & 0.98 & 0.97 & \dots \\ 0.98 & 0.99 & 1 & 0.99 & 0.98 & \dots \\ 0.97 & 0.98 & 0.99 & 1 & 0.99 & \dots \\ 0.96 & 0.97 & 0.98 & 0.99 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

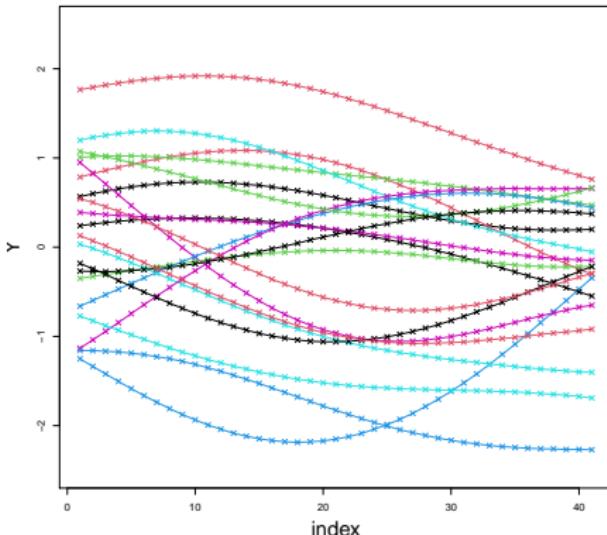


Each line is one sample.

We can think of Gaussian processes as an infinite dimensional distribution over functions - all we need to do is change the indexing

# Consider $d = 50$

$$\mu = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.99 & 0.98 & 0.97 & 0.96 & \dots \\ 0.99 & 1 & 0.99 & 0.98 & 0.97 & \dots \\ 0.98 & 0.99 & 1 & 0.99 & 0.98 & \dots \\ 0.97 & 0.98 & 0.99 & 1 & 0.99 & \dots \\ 0.96 & 0.97 & 0.98 & 0.99 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$



Each line is one sample.

We can think of Gaussian processes as an infinite dimensional distribution over functions - all we need to do is change the indexing

# Contents

Univariate and multivariate Gaussian distributions

Gaussian processes

Connection to Bayesian Linear regression

Resources

Summary

# Gaussian processes

- A stochastic process is a collection of random variables indexed by some variable  $x \in \mathcal{X}$

$$y = \{y(x) : x \in \mathcal{X}\}$$

- Usually  $y(x) \in \mathbb{R}$  and  $\mathcal{X} \subset \mathbb{R}^n$  - think of  $y$  as a function of  $x$ .
- If  $|\mathcal{X}| = \infty$ ,  $y$  is an infinite dimensional process.

# Gaussian processes

- A stochastic process is a collection of random variables indexed by some variable  $x \in \mathcal{X}$

$$y = \{y(x) : x \in \mathcal{X}\}$$

- Usually  $y(x) \in \mathbb{R}$  and  $\mathcal{X} \subset \mathbb{R}^n$  - think of  $y$  as a function of  $x$ .
- If  $|\mathcal{X}| = \infty$ ,  $y$  is an infinite dimensional process.

# Gaussian processes

- A stochastic process is a collection of random variables indexed by some variable  $x \in \mathcal{X}$

$$y = \{y(x) : x \in \mathcal{X}\}$$

- Usually  $y(x) \in \mathbb{R}$  and  $\mathcal{X} \subset \mathbb{R}^n$  - think of  $y$  as a function of  $x$ .
- If  $|\mathcal{X}| = \infty$ ,  $y$  is an infinite dimensional process.

# Gaussian processes

- A stochastic process is a collection of random variables indexed by some variable  $x \in \mathcal{X}$

$$y = \{y(x) : x \in \mathcal{X}\}$$

- Usually  $y(x) \in \mathbb{R}$  and  $\mathcal{X} \subset \mathbb{R}^n$  - think of  $y$  as a function of  $x$ .
- If  $|\mathcal{X}| = \infty$ ,  $y$  is an infinite dimensional process.

# Gaussian processes

- Thankfully, to understand the law of  $y$  we only need consider the finite dimensional distributions (FDDs), i.e., for all  $x_1, \dots, x_n$  and for all  $n \in \mathbb{N}$

$$\mathbb{P}(y(x_1) \leq c_1, \dots, y(x_n) \leq c_n)$$

as these uniquely determine the law of  $y$ .

- A **Gaussian process** is a stochastic process with Gaussian FDDs, i.e.,

$$(y(x_1), \dots, y(x_n)) \sim N_n(\mu, \Sigma)$$

Write  $y(\cdot) \sim GP$  to denote that the *function*  $y$  is a GP.

# Gaussian processes

- Thankfully, to understand the law of  $y$  we only need consider the finite dimensional distributions (FDDs), i.e., for all  $x_1, \dots, x_n$  and for all  $n \in \mathbb{N}$

$$\mathbb{P}(y(x_1) \leq c_1, \dots, y(x_n) \leq c_n)$$

as these uniquely determine the law of  $y$ .

- A **Gaussian process** is a stochastic process with Gaussian FDDs, i.e.,

$$(y(x_1), \dots, y(x_n)) \sim N_n(\mu, \Sigma)$$

Write  $y(\cdot) \sim GP$  to denote that the *function*  $y$  is a GP.

# Gaussian processes

- Thankfully, to understand the law of  $y$  we only need consider the finite dimensional distributions (FDDs), i.e., for all  $x_1, \dots, x_n$  and for all  $n \in \mathbb{N}$

$$\mathbb{P}(y(x_1) \leq c_1, \dots, y(x_n) \leq c_n)$$

as these uniquely determine the law of  $y$ .

- A **Gaussian process** is a stochastic process with Gaussian FDDs, i.e.,

$$(y(x_1), \dots, y(x_n)) \sim N_n(\mu, \Sigma)$$

Write  $y(\cdot) \sim GP$  to denote that the *function*  $y$  is a GP.

# Mean and covariance function

- To fully specify the law of a Gaussian *distribution* we only need the mean and variance.

$$Y \sim N(\mu, \Sigma)$$

- To fully specify the law of a Gaussian *process*, we need to specify mean and covariance **functions**.

$$y(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$$

where

$$\begin{aligned}\mathbb{E}(y(x)) &= m(x) \\ \text{Cov}(y(x), y(x')) &= k(x, x')\end{aligned}$$

# Mean and covariance function

- To fully specify the law of a Gaussian *distribution* we only need the mean and variance.

$$Y \sim N(\mu, \Sigma)$$

- To fully specify the law of a Gaussian *process*, we need to specify mean and covariance **functions**.

$$y(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$$

where

$$\begin{aligned}\mathbb{E}(y(x)) &= m(x) \\ \text{Cov}(y(x), y(x')) &= k(x, x')\end{aligned}$$

# Mean and covariance function

- To fully specify the law of a Gaussian *distribution* we only need the mean and variance.

$$Y \sim N(\mu, \Sigma)$$

- To fully specify the law of a Gaussian *process*, we need to specify mean and covariance **functions**.

$$y(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$$

where

$$\begin{aligned}\mathbb{E}(y(x)) &= m(x) \\ \text{Cov}(y(x), y(x')) &= k(x, x')\end{aligned}$$

# Specifying the mean function

- We can use any mean function we want  $m(x) = \mathbb{E}(y(x))$
- Most popular choices are  $m(x) = 0$  or  $m(x) = \text{const}$  for all  $x$ , or  $m(x) = \beta^\top x$ .
- Using a neural network is another popular choice.

# Specifying the mean function

- We can use any mean function we want  $m(x) = \mathbb{E}(y(x))$
- Most popular choices are  $m(x) = 0$  or  $m(x) = \text{const}$  for all  $x$ , or  $m(x) = \beta^\top x$ .
- Using a neural network is another popular choice.

# Specifying the mean function

- We can use any mean function we want  $m(x) = \mathbb{E}(y(x))$
- Most popular choices are  $m(x) = 0$  or  $m(x) = \text{const}$  for all  $x$ , or  $m(x) = \beta^\top x$ .
- Using a neural network is another popular choice.

# Covariance functions

- We usually use a covariance function that is a function of the indexes/locations

$$k(x, x') = \text{Cov}(y(x), y(x')),$$

$k$  must be a positive semi-definite function, i.e., lead to valid covariance matrices.

- Given locations  $x_1, \dots, x_n$ , the  $n \times n$  Gram matrix  $K$  with  $K_{ij} = k(x_i, x_j)$  must be a positive semi-definite matrix.
- A matrix  $K$  is positive semi-definite if for any vector  $u$ ,  $u^\top Ku \geq 0$ .

# Covariance functions

- We usually use a covariance function that is a function of the indexes/locations

$$k(x, x') = \text{Cov}(y(x), y(x')),$$

$k$  must be a positive semi-definite function, i.e., lead to valid covariance matrices.

- Given locations  $x_1, \dots, x_n$ , the  $n \times n$  Gram matrix  $K$  with  $K_{ij} = k(x_i, x_j)$  must be a positive semi-definite matrix.
- A matrix  $K$  is positive semi-definite if for any vector  $u$ ,  $u^\top K u \geq 0$ .

# Covariance functions

- We often assume  $k$  is a function of only the distance between locations

$$\text{Cov}(y(x), y(x')) = k(x - x')$$

which results in a **stationary** processes.

- If  $\text{Cov}(y(x), y(x')) = k(||x - x'||)$  the covariance function is said to be **isotropic**.
- The covariance function determines the *nature* of the GP.
- $k$  determines the hypothesis space/space of functions

# Covariance functions

- We often assume  $k$  is a function of only the distance between locations

$$\text{Cov}(y(x), y(x')) = k(x - x')$$

which results in a **stationary** processes.

- If  $\text{Cov}(y(x), y(x')) = k(||x - x'||)$  the covariance function is said to be **isotropic**.
- The covariance function determines the *nature* of the GP.
- $k$  determines the hypothesis space/space of functions

# Covariance functions

- We often assume  $k$  is a function of only the distance between locations

$$\text{Cov}(y(x), y(x')) = k(x - x')$$

which results in a **stationary** processes.

- If  $\text{Cov}(y(x), y(x')) = k(||x - x'||)$  the covariance function is said to be **isotropic**.
- The covariance function determines the *nature* of the GP.
- $k$  determines the hypothesis space/space of functions

# Covariance functions

- We often assume  $k$  is a function of only the distance between locations

$$\text{Cov}(y(x), y(x')) = k(x - x')$$

which results in a **stationary** processes.

- If  $\text{Cov}(y(x), y(x')) = k(||x - x'||)$  the covariance function is said to be **isotropic**.
- The covariance function determines the *nature* of the GP.
- $k$  determines the hypothesis space/space of functions

## How do we draw samples from a GP?

- Given the mean function and covariance function for a GP, we can draw samples using a multivariate Gaussian distribution.
- To sample from the multivariate Gaussian distribution, we need a mean vector and a covariance matrix.
- The mean vector is obtained from the mean function.
- The covariance matrix is obtained from the covariance function.

## Sampling from a GP

- RBF/Squared-exponential/exponentiated quadratic

$$k(x, x') = s_f \exp\left(-\frac{(x - x')^2}{2\ell^2}\right),$$

where  $s_f$  is the variance parameter and  $\ell$  the length-scale parameter.

- If  $s_f = 1$  and  $\ell = 1$ , we get

$$k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)$$

- Say we have a vector of  $x$  values, like

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^\top.$$

- These are the indexes of the stochastic process.

## Sampling from a GP

- We now compute the covariance matrix

$$K_{XX} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

- We assume the mean function is constant and equal to zero,  $m(x) = 0$ .
- To generate functions from this GP, we will then sample from

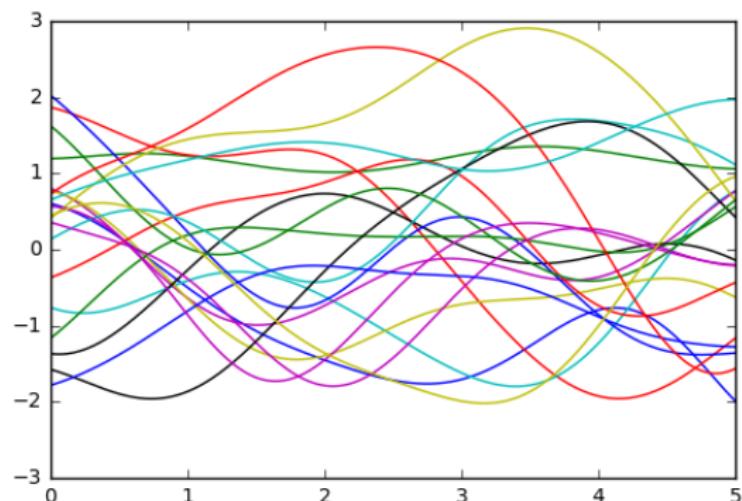
$$\mathbf{y} \sim N_n \left( \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \right)$$

- What we plot is  $\mathbf{x}$  and  $\mathbf{y}$ .

# Examples

RBF/Squared-exponential/exponentiated quadratic

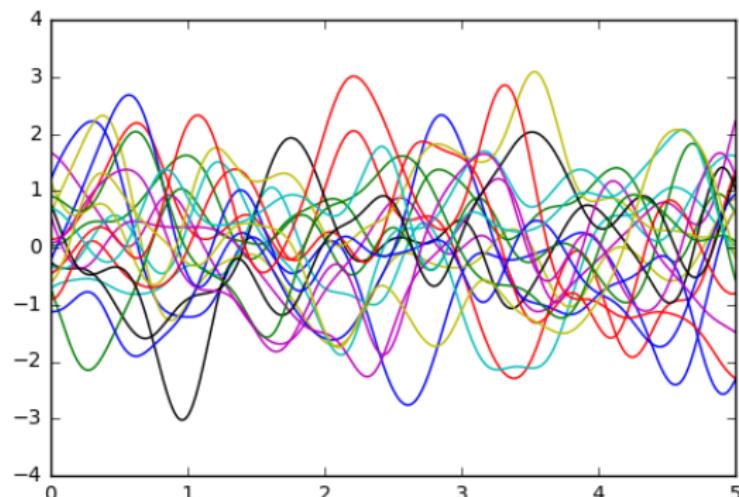
$$k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)$$



# Examples

RBF/Squared-exponential/exponentiated quadratic

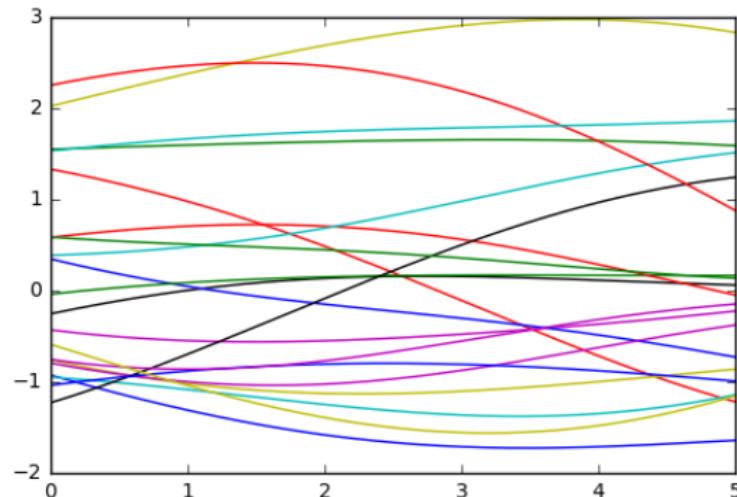
$$k(x, x') = \exp\left(-\frac{1}{2} \frac{(x - x')^2}{0.25^2}\right)$$



# Examples

RBF/Squared-exponential/exponentiated quadratic

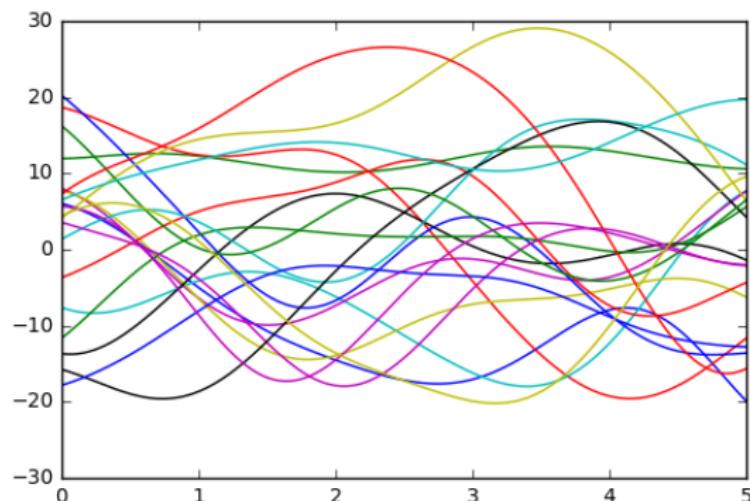
$$k(x, x') = \exp\left(-\frac{1}{2} \frac{(x - x')^2}{4^2}\right)$$



# Examples

RBF/Squared-exponential/exponentiated quadratic

$$k(x, x') = \textcolor{red}{100} \exp\left(-\frac{1}{2}(x - x')^2\right)$$



# Examples

Matérn covariance function

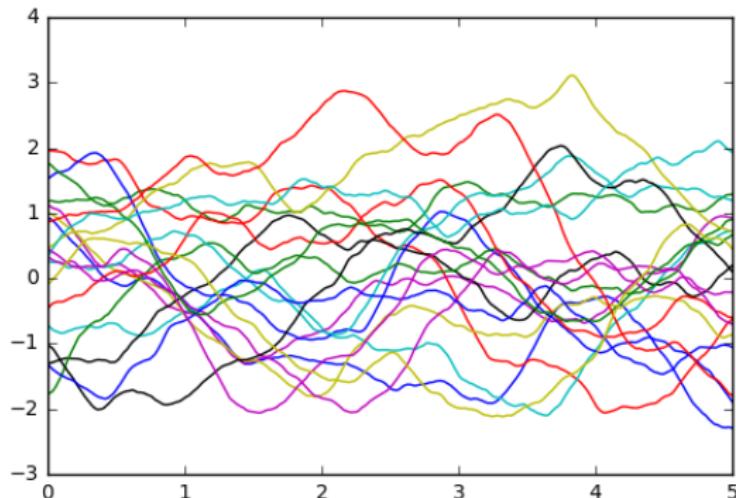
$$k(x, x') = s_f \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu}(x - x')}{\ell} \right)^\nu K_\nu \left( \frac{\sqrt{2\nu}(x - x')}{\ell} \right)$$
$$k(x, x') = s_f \left( 1 + \frac{\sqrt{3}|x - x'|}{\ell} \right) \exp \left( -\frac{\sqrt{3}|x - x'|}{\ell} \right), \quad \nu = \frac{3}{2},$$

where  $K_\nu(\cdot)$  is the modified Bessel function of the second kind.

# Examples

## Matérn 3/2

$$k(x, x') \sim (1 + |x - x'|) \exp(-|x - x'|)$$



# Examples

Many other covariance functions: constant, linear, polynomial, exponential, etc.

## Building new kernels (I)

Given two valid kernels  $k_1(\mathbf{x}, \mathbf{x}')$  and  $k_2(\mathbf{x}, \mathbf{x}')$ , the following *new* kernels are also valid kernels

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

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

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

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

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

where  $c > 0$  is a constant,  $f(\cdot)$  is any function, and  $q(\cdot)$  is a polynomial with non-negative coefficients.

## Building new kernels (II)

Given two valid kernels  $k_1(\mathbf{x}, \mathbf{x}')$  and  $k_2(\mathbf{x}, \mathbf{x}')$ , the following *new* kernels are also valid kernels

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

$$k(\mathbf{x}, \mathbf{x}') = k_3(\phi(\mathbf{x}), \phi(\mathbf{x}'))$$

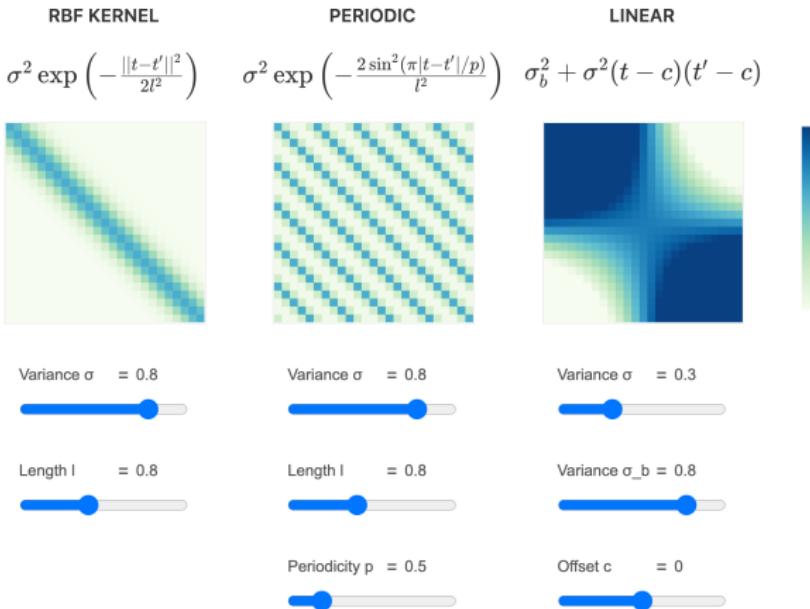
$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A} \mathbf{x}'$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}_b)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a)k_b(\mathbf{x}_b, \mathbf{x}_b),$$

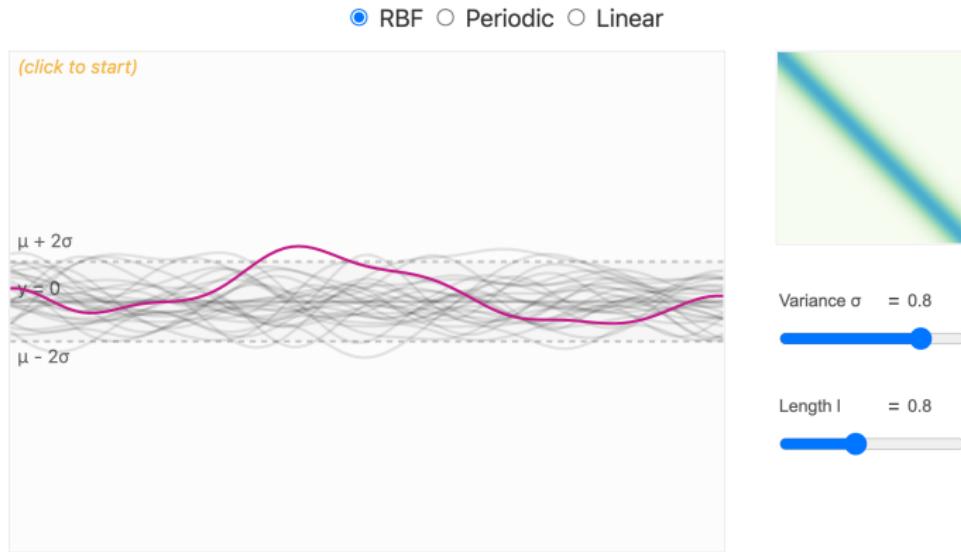
where  $\phi(\cdot)$  maps as  $\mathbb{R}^D \rightarrow \mathbb{R}^N$ ,  $k_3(\cdot, \cdot)$  is a valid kernel in  $\mathbb{R}^N$ ,  $\mathbf{A}$  is a symmetric positive semidefinite matrix,  $\mathbf{x}_a$  and  $\mathbf{x}_b$  are vectors such that  $\mathbf{x} = [\mathbf{x}_a^\top, \mathbf{x}_b^\top]^\top$ , and  $k_a(\cdot, \cdot)$  and  $k_b(\cdot, \cdot)$  are valid kernels over their respective spaces.

# Visual exploration: covariance matrices



This figure shows various kernels that can be used with Gaussian processes. Each kernel has different parameters, which can be changed by adjusting the according sliders. When grabbing a slider, information on how the current parameter influences the kernel will be shown on the right.

# Visual exploration: samples from GPs



Clicking on the graph results in continuous **samples** drawn from a Gaussian process using the selected kernel. After each draw, the previous sample fades into the background. Over time, it is possible to see that functions are distributed normally around the mean  $\mu$ .

Taken from: "Visual exploration of Gaussian processes" by J Götler, R Kehlbeck and O Deussen (2019)

# Why use Gaussian processes?

- Why would we want to use this very restricted class of model?
- Gaussian **distributions** have several properties that make them easy to work with: sums of Gaussians are Gaussian, and marginal distributions of multivariate Gaussians are still Gaussian.

# Conditional distributions are still Gaussian

Suppose

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2(\mu, \Sigma)$$

where

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

Then

$$Y_2 | Y_1 = y_1 \sim N\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$$

# Conditional distributions are still Gaussian

Suppose

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2(\mu, \Sigma)$$

where

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

Then

$$Y_2 | Y_1 = y_1 \sim N\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$$

# Conditional updates of Gaussian processes

Suppose  $f$  is a Gaussian process, then

$$f(x_1), \dots, f(x_n), f(x) \sim N_{n+1}(0, \Sigma)$$

where

$$\begin{aligned}\Sigma &= \left( \begin{array}{ccc|c} k(x_1, x_1) & \dots & k(x_1, x_n) & k(x_1, x) \\ \vdots & & \vdots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) & k(x_n, x) \\ \hline k(x, x_1) & \dots & k(x, x_n) & k(x, x) \end{array} \right) \\ &= \left( \begin{array}{c|c} K_{XX} & k_X(x) \\ \hline k_X(x)^T & k(x, x) \end{array} \right)\end{aligned}$$

where  $X = \{x_1, \dots, x_n\}$ ,  $[K_{XX}]_{ij} = k(x_i, x_j)$  is the Gram/kernel matrix, and  $[k_X(x)]_j = k(x_j, x)$

# Conditional updates of Gaussian processes

Suppose  $f$  is a Gaussian process, then

$$f(x_1), \dots, f(x_n), f(x) \sim N_{n+1}(0, \Sigma)$$

where

$$\begin{aligned}\Sigma &= \left( \begin{array}{ccc|c} k(x_1, x_1) & \dots & k(x_1, x_n) & k(x_1, x) \\ \vdots & & \vdots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) & k(x_n, x) \\ \hline k(x, x_1) & \dots & k(x, x_n) & k(x, x) \end{array} \right) \\ &= \left( \begin{array}{c|c} K_{XX} & k_X(x) \\ \hline k_X(x)^T & k(x, x) \end{array} \right)\end{aligned}$$

where  $X = \{x_1, \dots, x_n\}$ ,  $[K_{XX}]_{ij} = k(x_i, x_j)$  is the Gram/kernel matrix, and  $[k_X(x)]_j = k(x_j, x)$

# Conditional updates of Gaussian processes

Then

$$f(x) | f(x_1), \dots, f(x_n) \sim N(\bar{m}(x), \bar{k}(x))$$

where

$$\bar{m}(x) = k_X(x)^\top K_{XX}^{-1} \mathbf{f}$$

with

$$\mathbf{f} = (f(x_1), \dots, f(x_n))^\top$$

$$k_X(x)^\top = (k(x, x_1) \ k(x, x_2) \ \dots \ k(x, x_n)) \in \mathbb{R}^{1 \times n}$$

and

$$\bar{k}(x) = k(x, x) - k_X(x)^\top K_{XX}^{-1} k_X(x)$$

What this means in practice is that if we know  $\mathbf{f}$ , we can use it to predict  $f(x)$  as a Gaussian distribution with mean  $\bar{m}(x)$  and variance  $\bar{k}(x)$ .

# Conditional updates of Gaussian processes

Then

$$f(x) | f(x_1), \dots, f(x_n) \sim N(\bar{m}(x), \bar{k}(x))$$

where

$$\bar{m}(x) = k_X(x)^\top K_{XX}^{-1} \mathbf{f}$$

with

$$\mathbf{f} = (f(x_1), \dots, f(x_n))^\top$$

$$k_X(x)^\top = (k(x, x_1) \ k(x, x_2) \ \dots \ k(x, x_n)) \in \mathbb{R}^{1 \times n}$$

and

$$\bar{k}(x) = k(x, x) - k_X(x)^\top K_{XX}^{-1} k_X(x)$$

What this means in practice is that if we know  $\mathbf{f}$ , we can use it to predict  $f(x)$  as a Gaussian distribution with mean  $\bar{m}(x)$  and variance  $\bar{k}(x)$ .

# Conditional updates of Gaussian processes

Then

$$f(x) | f(x_1), \dots, f(x_n) \sim N(\bar{m}(x), \bar{k}(x))$$

where

$$\bar{m}(x) = k_X(x)^\top K_{XX}^{-1} \mathbf{f}$$

with

$$\mathbf{f} = (f(x_1), \dots, f(x_n))^\top$$

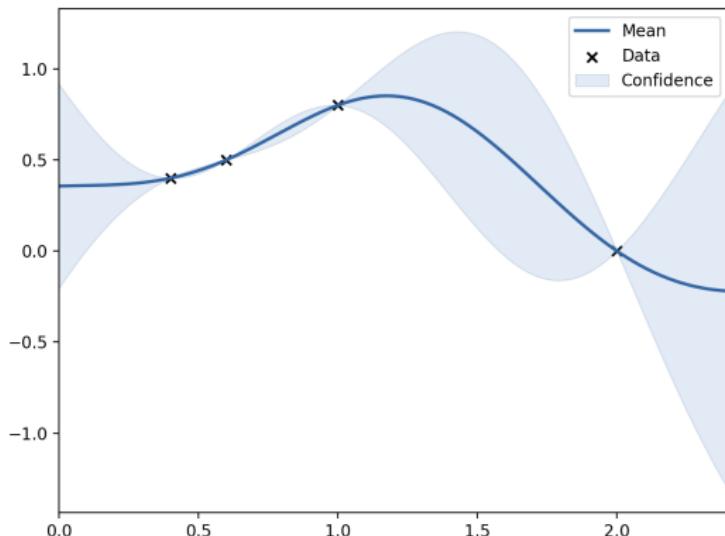
$$k_X(x)^\top = (k(x, x_1) \ k(x, x_2) \ \dots \ k(x, x_n)) \in \mathbb{R}^{1 \times n}$$

and

$$\bar{k}(x) = k(x, x) - k_X(x)^\top K_{XX}^{-1} k_X(x)$$

What this means in practice is that if we know  $\mathbf{f}$ , we can use it to predict  $f(x)$  as a Gaussian distribution with mean  $\bar{m}(x)$  and variance  $\bar{k}(x)$ .

# Interpolation



Solid line  $\bar{m}(x) = k_X^\top(x) K_{XX}^{-1} \mathbf{f}$

Shaded region  $\bar{m}(x) \pm 1.96 \sqrt{\bar{k}(x)}$

$$\bar{k}(x) = k(x, x) - k_X^\top(x) K_{XX}^{-1} k_X(x)$$

## Noisy observations - Regression

- In practice, we don't usually observe  $f(x)$  directly.
- If we observe

$$y_i = f(x_i) + \epsilon,$$

where  $\epsilon \sim N(0, \sigma^2)$  then

$$y_1, \dots, y_n, f(x) \sim N_{n+1}(0, \Sigma)$$

where

$$\Sigma = \left( \begin{array}{cc|c} K_{XX} + \sigma^2 I & & \begin{matrix} k(x_1, x) \\ k(x_2, x) \\ \vdots \\ k(x_n, x) \end{matrix} \\ \hline k(x, x_1) & k(x, x_2) & \dots & k(x, x_n) & k(x, x) \end{array} \right)$$

# Noisy observations - Regression

- In this way

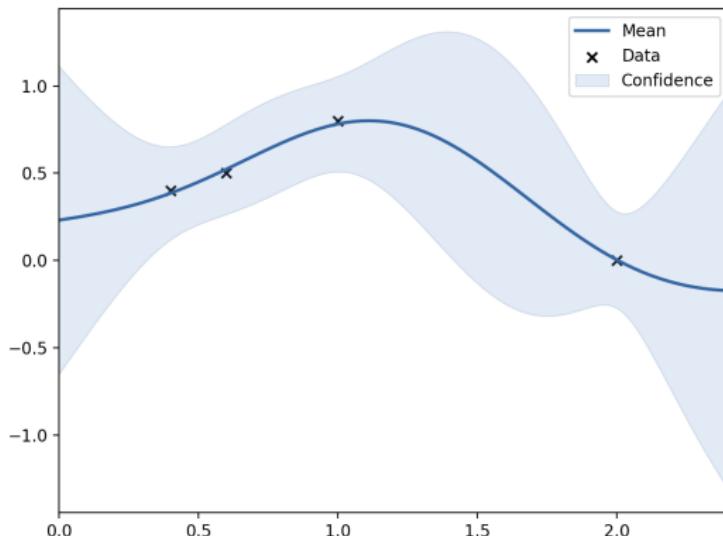
$$f(x) \mid y_1, \dots, y_n \sim N(\bar{m}(x), \bar{k}(x))$$

where

$$\bar{m}(x) = k_X(x)^\top (K_{XX} + \sigma^2 I)^{-1} \mathbf{y}$$

$$\bar{k}(x) = k(x, x) - k_X(x)^\top (K_{XX} + \sigma^2 I)^{-1} k_X(x)$$

# Noise standard deviation $\sigma = 0.1$

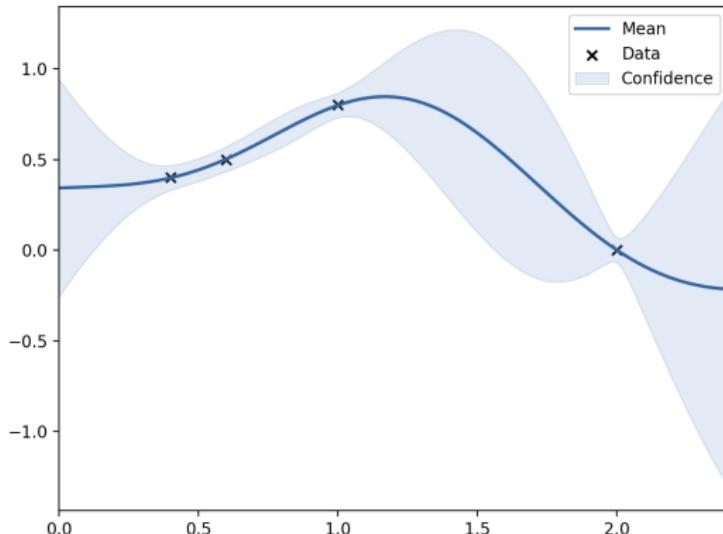


Solid line       $\bar{m}(x) = k_X(x)^\top K_{XX}^{-1} \mathbf{y}$

Shaded region       $\bar{m}(x) \pm 1.96 \sqrt{\bar{k}(x)}$

$$\bar{k}(x) = k(x, x) - k_X(x)^\top (K_{XX}^{-1} + \sigma^2 I) k_X(x)$$

# Noise standard deviation $\sigma = 0.025$

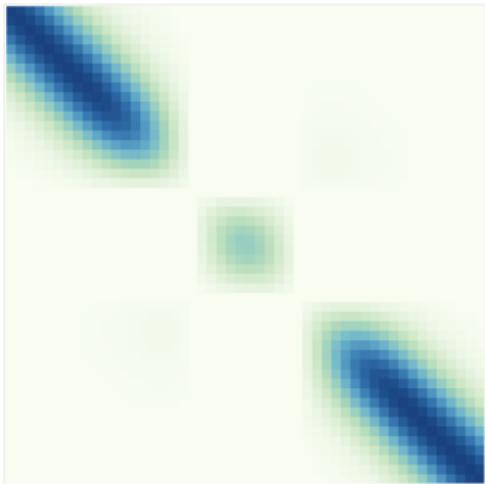
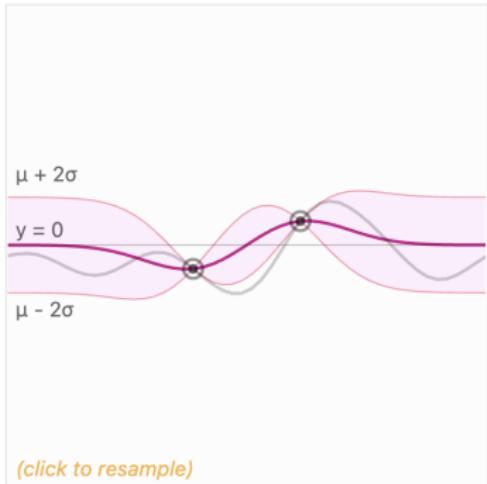


Solid line       $\bar{m}(x) = k_X(x)^\top K_{XX}^{-1} \mathbf{y}$

Shaded region       $\bar{m}(x) \pm 1.96 \sqrt{\bar{k}(x)}$

$$\bar{k}(x) = k(x, x) - k_X(x)^\top (K_{XX}^{-1} + \sigma^2 I) k_X(x)$$

# Visual exploration



Taken from: "Visual exploration of Gaussian processes" by J Götler, R Kehlbeck and O Deussen (2019)

## Practical aspects

- ❑ If we knew the covariance function we should use, GPs work great!
- ❑ Unfortunately, we don't usually know this.
- ❑ We pick a covariance function from a small set, based usually on differentiability considerations.

## Practical aspects

- ❑ If we knew the covariance function we should use, GPs work great!
- ❑ Unfortunately, we don't usually know this.
- ❑ We pick a covariance function from a small set, based usually on differentiability considerations.

## Practical aspects

- If we knew the covariance function we should use, GPs work great!
- Unfortunately, we don't usually know this.
- We pick a covariance function from a small set, based usually on differentiability considerations.

## Practical aspects

- ❑ Possibly try a few (plus combinations of a few) covariance functions, and attempt to make a good choice using some sort of empirical evaluation.
- ❑ Covariance functions often contain hyper-parameters. E.g RBF kernel

$$k(x, x') = s_f^2 \exp\left(-\frac{1}{2} \frac{(x - x')^2}{\ell^2}\right)$$

Estimate these using your favourite statistical procedure (maximum likelihood, cross-validation, Bayes, expert judgement etc)

## Practical aspects

- ❑ Possibly try a few (plus combinations of a few) covariance functions, and attempt to make a good choice using some sort of empirical evaluation.
- ❑ Covariance functions often contain hyper-parameters. E.g RBF kernel

$$k(x, x') = s_f^2 \exp\left(-\frac{1}{2} \frac{(x - x')^2}{\ell^2}\right)$$

Estimate these using your favourite statistical procedure (maximum likelihood, cross-validation, Bayes, expert judgement etc)

## Marginal likelihood

- A popular way to estimate the hyperparameters of the covariance function is through maximizing the logarithm of the marginal likelihood.
- The logarithm of the marginal likelihood is given as

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\mathbf{y}^\top(K_{xx} + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K_{xx} + \sigma^2 I| - \frac{n}{2}\log 2\pi.$$

- If we know  $\mathbf{x}$  and  $\mathbf{y}$ , the only unknowns in  $\log p(\mathbf{y}|\mathbf{x})$  are the kernel hyperparameters, e.g.  $s_f$  and  $\ell$ , and the parameter  $\sigma$ .
- We can then optimise  $\log p(\mathbf{y}|\mathbf{x})$  wrt these parameters using a gradient-descent like procedure.

## Marginal likelihood

- A popular way to estimate the hyperparameters of the covariance function is through maximizing the logarithm of the marginal likelihood.
- The logarithm of the marginal likelihood is given as

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\mathbf{y}^\top(K_{XX} + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K_{XX} + \sigma^2 I| - \frac{n}{2}\log 2\pi.$$

- If we know  $\mathbf{x}$  and  $\mathbf{y}$ , the only unknowns in  $\log p(\mathbf{y}|\mathbf{x})$  are the kernel hyperparameters, e.g.  $s_f$  and  $\ell$ , and the parameter  $\sigma$ .
- We can then optimise  $\log p(\mathbf{y}|\mathbf{x})$  wrt these parameters using a gradient-descent like procedure.

## Marginal likelihood

- A popular way to estimate the hyperparameters of the covariance function is through maximizing the logarithm of the marginal likelihood.
- The logarithm of the marginal likelihood is given as

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\mathbf{y}^\top(K_{XX} + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K_{XX} + \sigma^2 I| - \frac{n}{2}\log 2\pi.$$

- If we know  $\mathbf{x}$  and  $\mathbf{y}$ , the only unknowns in  $\log p(\mathbf{y}|\mathbf{x})$  are the kernel hyperparameters, e.g.  $s_f$  and  $\ell$ , and the parameter  $\sigma$ .
- We can then optimise  $\log p(\mathbf{y}|\mathbf{x})$  wrt these parameters using a gradient-descent like procedure.

# Log-marginal likelihood surface ( $\sigma$ and $\ell$ )

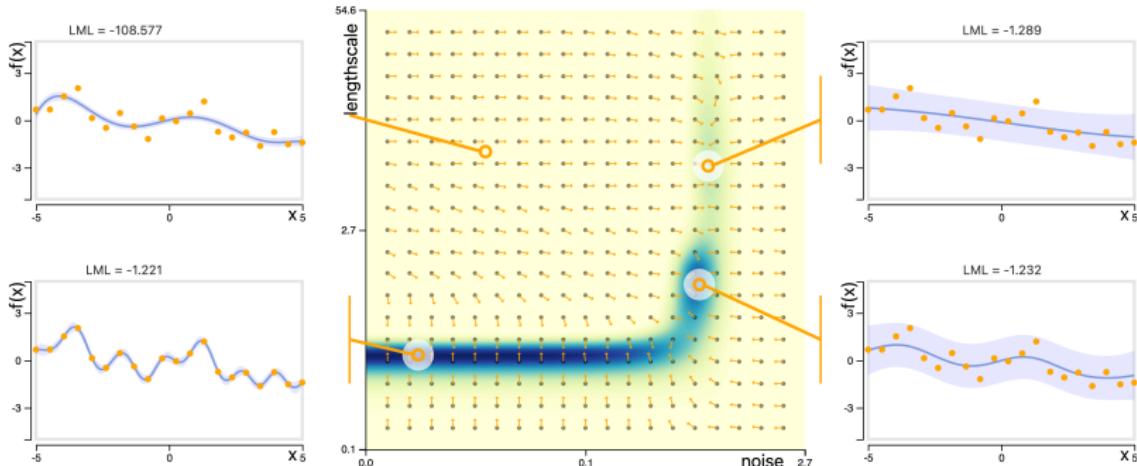


FIGURE 5: Training data (orange discs), log-marginal likelihood contour and three possible GP fits (left-bottom and right-hand panels) corresponding to the three local optima at  $(\sigma_n, \ell) = (0.02, 0.36), (0.97, 5.80), (0.76, 1.13)$ , respectively. All three hyperparameter settings are valid, therefore, the corresponding GP models make sense: While the bottom left GP model explains the (noisy) data well, the top right GP fit explains the data by a near-linear function (long lengthscale) and an high noise level. The global optimum (bottom right) has a slightly better log-marginal likelihood value and is a compromise between the other two other local optima, discovering the latent sinusoidal wave that generated the data while accounting for a fairly high level of measurement noise.

Taken from: "A Practical Guide to Gaussian Processes" by M. Deisenroth, Y. Luo and M. van der Wilk (2013)

# Computational cost

- ❑ One difficulty with GPs is the computational cost of training them:  $O(n^3)$  (and  $O(n^2)$  memory).
- ❑ They work our of the box for  $n$  in the order of a few thousands.
- ❑ There are many ways to side-step this cost: inducing inputs, efficient matrix-vector multiplications, random features, etc.
- ❑ These days we can use GPs for  $n$  in the order of tens of millions.

# Computational cost

- ❑ One difficulty with GPs is the computational cost of training them:  $O(n^3)$  (and  $O(n^2)$  memory).
- ❑ They work our of the box for  $n$  in the order of a few thousands.
- ❑ There are many ways to side-step this cost: inducing inputs, efficient matrix-vector multiplications, random features, etc.
- ❑ These days we can use GPs for  $n$  in the order of tens of millions.

# Computational cost

- ❑ One difficulty with GPs is the computational cost of training them:  $O(n^3)$  (and  $O(n^2)$  memory).
- ❑ They work our of the box for  $n$  in the order of a few thousands.
- ❑ There are many ways to side-step this cost: inducing inputs, efficient matrix-vector multiplications, random features, etc.
- ❑ These days we can use GPs for  $n$  in the order of tens of millions.

## Computational cost

- ❑ One difficulty with GPs is the computational cost of training them:  $O(n^3)$  (and  $O(n^2)$  memory).
- ❑ They work our of the box for  $n$  in the order of a few thousands.
- ❑ There are many ways to side-step this cost: inducing inputs, efficient matrix-vector multiplications, random features, etc.
- ❑ These days we can use GPs for  $n$  in the order of tens of millions.

# Contents

Univariate and multivariate Gaussian distributions

Gaussian processes

Connection to Bayesian Linear regression

Resources

Summary

# Weight-space view of GPs

- The way we introduced GPs before is known as the *function-space* view.
- Another way to introduce GPs is through Bayesian linear regression, the *weight-space* view.

## Linear model (I)

- Say we have a training set  $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, \dots, n\}$ .
- $\mathbf{x}_i \in \mathbb{R}^D$ ,  $y_i \in \mathbb{R}$ .
- We have a design matrix  $\mathbf{X} \in \mathbb{R}^{n \times D}$ , and an output vector  $\mathbf{y}$ ,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}, \quad \mathbf{y} = [y_1 \quad y_2 \quad \cdots \quad y_n]^\top.$$

- Therefore  $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ .

## Linear model (II)

- The standard linear model assumes

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}, \quad y = f(\mathbf{x}) + \epsilon,$$

where  $\mathbf{w} \in \mathbb{R}^D$  is a parameter vector,  $y$  is the corresponding observation for  $\mathbf{x}$ , and  $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ .

- We assume **iid** observations.
- The likelihood for this model,  $p(\mathbf{y}|\mathbf{X}, \mathbf{w})$ , follows as

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \prod_{i=1}^n p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma_n^2}\right] \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2} |\mathbf{y} - \mathbf{X}\mathbf{w}|^2\right) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma_n^2 \mathbf{I}). \end{aligned}$$

## Linear model (III)

- In **Bayesian linear regression**, we specify a prior distribution over  $\mathbf{w}$ , for example,

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p),$$

where  $\Sigma_p$  is a covariance matrix.

- Bayes theorem,

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \quad p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})},$$

where

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}.$$

## Linear model (IV)

- For the linear model,

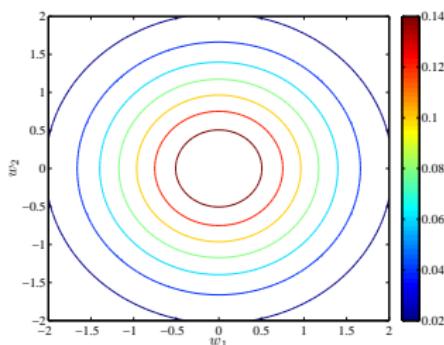
$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \mathcal{N}(\mathbf{w} \Big| \underbrace{\frac{1}{\sigma_n^2} \mathbf{A}^{-1} \mathbf{X}^\top \mathbf{y}, \mathbf{A}^{-1}}_{\hat{\mathbf{w}}}).$$

where  $\mathbf{A} = \sigma_n^{-2} \mathbf{X}^\top \mathbf{X} + \boldsymbol{\Sigma}_p^{-1}$  is a covariance matrix.

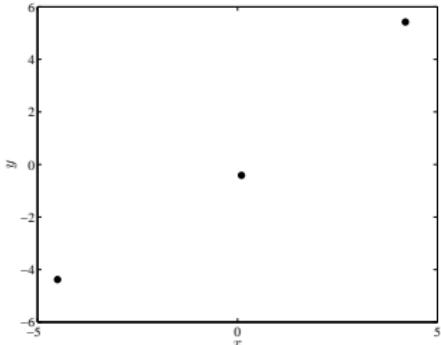
- The predictive distribution for  $f_* \equiv f(\mathbf{x}_*)$  at  $\mathbf{x}_*$  is given as

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{y}, \mathbf{X}) d\mathbf{w}, \\ &= \mathcal{N}\left(f_* \Big| \frac{1}{\sigma_n^2} \mathbf{x}_*^\top \mathbf{A}^{-1} \mathbf{X}^\top \mathbf{y}, \mathbf{x}_*^\top \mathbf{A}^{-1} \mathbf{x}_*\right). \end{aligned}$$

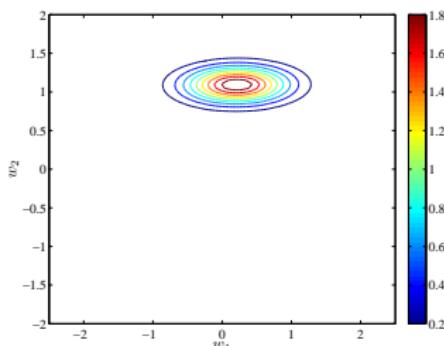
# Linear model (V)



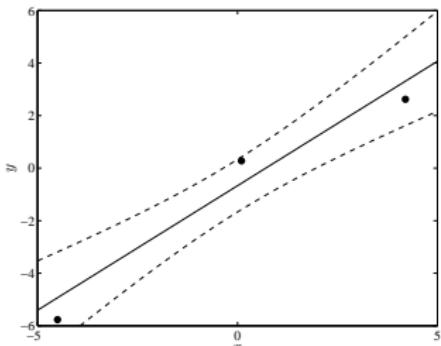
$p(\mathbf{w})$



$\mathcal{D} = (\mathbf{X}, \mathbf{y})$



$p(\mathbf{w}|\mathbf{y}, \mathbf{X})$



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

## Feature space (I)

- The Bayesian linear model is limited since it is linear in both  $\mathbf{x}$  and  $\mathbf{w}$ .
- We could use *basis functions* to introduce non-linearity in the model.

## Feature space (II)

- We introduce the function  $\phi(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^N$ .
- The new design matrix is  $\Phi(\mathbf{X}) \in \mathbb{R}^{n \times N}$ ,

$$\Phi(\mathbf{X}) = \begin{bmatrix} \phi(\mathbf{x}_1)^\top \\ \phi(\mathbf{x}_2)^\top \\ \vdots \\ \phi(\mathbf{x}_n)^\top \end{bmatrix},$$

- The model is now  $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x})$ , with  $\mathbf{w} \in \mathbb{R}^N$ .
- The equations remain the same simply changing  $\mathbf{X}$  for  $\Phi(\mathbf{X})$ .

## Feature space (III)

- The predictive distribution follows as

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} \left( f_* \middle| \frac{1}{\sigma_n^2} \phi(\mathbf{x}_*)^\top \mathbf{A}^{-1} \Phi^\top \mathbf{y}, \phi(\mathbf{x}_*)^\top \mathbf{A}^{-1} \phi(\mathbf{x}_*) \right),$$

where  $\Phi = \Phi(\mathbf{X})$ , and  $\mathbf{A} = \sigma_n^{-2} \Phi^\top \Phi + \Sigma_p^{-1}$ .

- Inverting  $\mathbf{A}$  is expensive for a large value of  $N$ .
- It can be shown that

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \mathcal{N} \left( f_* \middle| \phi_*^\top \Sigma_p \Phi^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \right. \\ &\quad \left. \phi_*^\top \Sigma_p \phi_* - \phi_*^\top \Sigma_p \Phi^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \Phi \Sigma_p \phi_* \right), \end{aligned}$$

where  $\phi(\mathbf{x}_*) = \phi_*$ ,  $\mathbf{y}^\top \mathbf{K} = \Phi \Sigma_p \Phi^\top$ .

- The feature space appears in the forms  $\phi_*^\top \Sigma_p \Phi^\top$ ,  $\phi_*^\top \Sigma_p \phi_*$ , and  $\Phi \Sigma_p \Phi^\top$ .

## Kernel trick

- Entries in the matrices appearing before can be written as  $\phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}')$ .
- The function  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}')$ , is the kernel or covariance function we introduced in the function space view of GPs.
- $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}') = \psi(\mathbf{x}) \cdot \psi(\mathbf{x}')$ , with  $\psi(\mathbf{x}) = \Sigma_p^{1/2} \phi(\mathbf{x}')$ .
- When a model only depends on inner products between vectors in the input space, those inner products can be replaced by  $k(\mathbf{x}, \mathbf{x}')$ .

## Going back to GPs

- For a GP, we specify the mean,  $m(\mathbf{x})$ , and the covariance function,  $k(\mathbf{x}, \mathbf{x}')$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))].$$

- In Bayesian linear regression, we have  $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x})$  with prior  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$ .
- If we compute  $\mathbb{E}[f(\mathbf{x})]$  and  $\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')]$ , we get

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0, \\ \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \phi(\mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}).$$

- Meaning that the Bayesian linear regression model is equivalent to a GP prior with mean zero and covariance  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}')$ .

# Contents

Univariate and multivariate Gaussian distributions

Gaussian processes

Connection to Bayesian Linear regression

Resources

Summary

# Gaussian process summer school



## Gaussian Process Summer School 2024

About Program Getting Started Labs Accommodation Registration

### About

The Gaussian Process and Uncertainty Quantification Summer School was held in Manchester, UK from **Monday, 9 September 2024 to Thursday, 12 September 2024** at the Kilburn Building, Lecture Theatre 1.1.

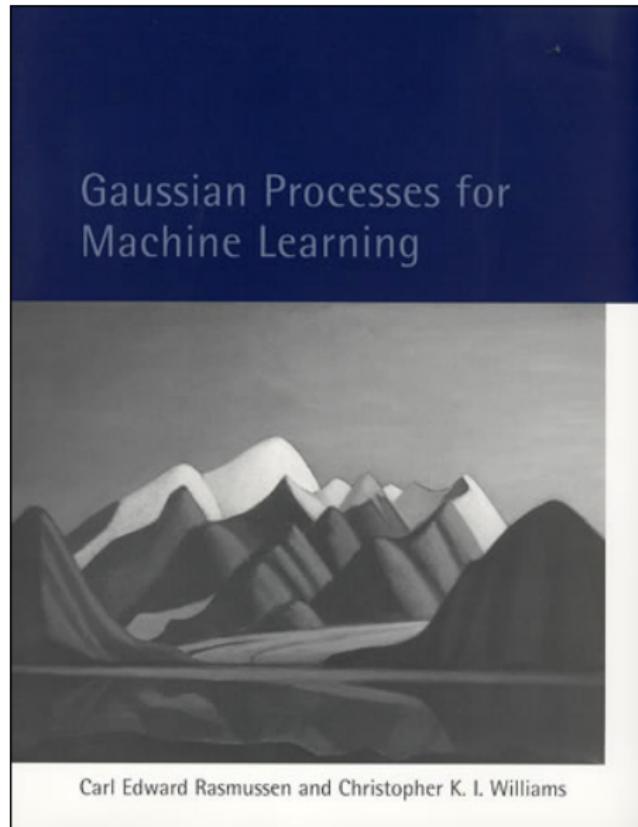
This webpage acts as a record of the program and talks given.

#### Organizers

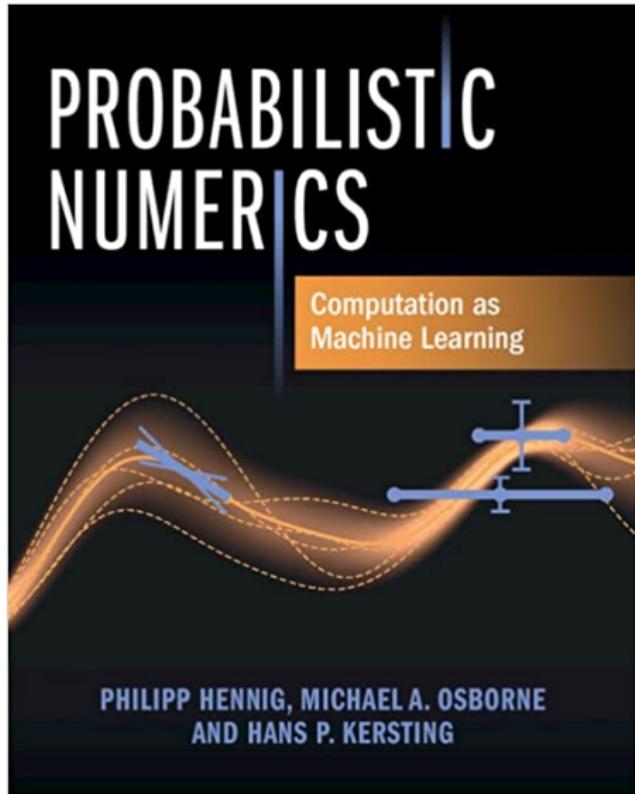
- Mauricio Alvarez, University of Manchester
- Carl H. Ek, University of Cambridge
- Richard Wilkinson, University of Nottingham
- Henry Moss, University of Cambridge
- Neil D. Lawrence, University of Cambridge

Run the second week of September, go to <http://gpss.cc>

# Book



# Probabilistic numerics



# Contents

Univariate and multivariate Gaussian distributions

Gaussian processes

Connection to Bayesian Linear regression

Resources

Summary

# Summary

- GPs are ubiquitous in statistics/ML.
- Popularity stems from
  - Naturalness of the framework
  - Mathematical tractability
  - Empirical success

# Acknowledgements

- Prof. Richard Wilkinson from the University of Nottingham for providing some of the material in these slides.